



ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES EXPERIMENTALS

MÁSTER EN INGENIERÍA INDUSTRIAL

Desarrollo de un sistema de bombeo de laboratorio a escala y del sistema de control para minimizar el coste energético

TRABAJO DE FIN DE MASTER

AUTOR:

Johnata Lorenzo Fernandes

DIRECTOR:

Roberto Sanchis Llopis

Castellón, Octubre de 2019

AGRADECIMIENTOS

Me gustaría agradecer en primer lugar a mis padres y seres queridos por creer en mí, apoyarme en todo momento y empujarme a conseguir mis metas.

También agradecerle a mi tutor Roberto Sanchis, por su compromiso, su paciencia y por guiarme durante todo este proyecto.

ÍNDICE

SIGLAS Y GLOSARIO	6
MEMORIA.....	11
PLANOS.....	111
PLIEGO DE CONDICIONES	135
PRESUPUESTO.....	141

SIGLAS Y GLOSARIO

PC	Personal computer
PLC	Programmable Logic Controller
NC	Normally closed – Normalmente cerrado
NO	Normally open – Normalmente abierto
PID	Controlador con parte proporcional, integral y derivativa
PI	Controlador con parte proporcional e integral únicamente
Kp	Ganancia estática
Ki	Ganancia integral
Ti	Tiempo integral
Nc	Número de combinaciones posibles
Nt	Número de tanques
Np	Número de bombas de llenado
Nv	Número de válvulas
ST	Structured Text – Texto estructurado
CFC	Continuous Function Chart
DI	Digital Input
DO	Digital Output
AI	Analog Input
AO	Analog Output
OPC	Open Platform Communications

ÍNDICE DE FIGURAS

Figura 1. Pozos (verde) y Tanques (rojo) de la maqueta.....	19
Figura 2. Bomba sumergible 24V.....	20
Figura 3.Caudalímetro.....	20
Figura 4. Electroválvula.....	21
Figura 5. Sensores de nivel capacitivos.....	21
Figura 6. PLC Wago y periféricos entrada-salida.....	22
Figura 7. Amplificador de potencia.....	23
Figura 8. Fuente de alimentación.....	23
Figura 9. Esquema hidráulico de la maqueta.....	24
Figura 10. Esquema eléctrico general.....	26
Figura 11. Bucle de control en lazo cerrado.....	29
Figura 12. Carga de datos - Herramienta Java de identificación.....	30
Figura 13. Identificación- Herramienta Java de identificación.....	31
Figura 14. Diseño del PID- Herramienta Java de diseño PID.....	32
Figura 15. Configuración hidráulica final.....	34
Figura 16.Comparativa curva de caudal estimado sin actualizar y caudal demandado real.....	45
Figura 17. Estructura de funcionamiento del código en Matlab.....	46
Figura 18. Inicializa - Bloque de código 1.....	47
Figura 19. Inicializa - Bloque de código 2.....	48
Figura 20. Inicializa - Bloque de código 3.....	48
Figura 21. Llama al opti - Bloque de código 1.....	48
Figura 22. Llama al opti - Bloque de código 2.....	49
Figura 23. Llama al opti - Bloque de código 3.....	49
Figura 24. Llama al opti - Bloque de código 4.....	50
Figura 25. Llama al opti - Bloque de código 5.....	50
Figura 26. Llama al opti - Bloque de código 6.....	51
Figura 27. Llama al opti - Bloque de código 7.....	51
Figura 28. Llama al opti - Bloque de código 8.....	51
Figura 29. Llama al opti - Bloque de código 9.....	52
Figura 30. Llama al opti - Bloque de código 10.....	52
Figura 31. Llama al opti - Bloque de código 11.....	52
Figura 32. Optimiza bombeo futuro - Bloque de código 2.....	53
Figura 33. Optimiza bombeo futuro - Bloque de código 3.....	53
Figura 34. Optimiza bombeo futuro - Bloque de código 4.....	53
Figura 35. Optimiza bombeo futuro - Bloque de código 5.....	54
Figura 36. Optimiza bombeo futuro - Bloque de código 6.....	54
Figura 37. Optimiza bombeo futuro - Bloque de código 7.....	54
Figura 38. Optimiza bombeo futuro - Bloque de código 8.....	55
Figura 39. Optimiza bombeo futuro - Bloque de código 9.....	55
Figura 40. Optimiza bombeo futuro - Bloque de código 10.....	56
Figura 41. Optimiza bombeo futuro - Bloque de código 11.....	56
Figura 42.Timer Lab - Bloque de código 1.....	56
Figura 43. Configuración de tareas PLC.....	57
Figura 44. Demanda Caudal - Bloque de código 1.....	58

Figura 45. Demanda Caudal - Bloque de código 2.....	58
Figura 46. Demanda Caudal - Bloque de código 3.....	59
Figura 47. Control Bombas - Bloque de código 1	61
Figura 48. Resultados del caso experimental 1	64
Figura 49. Resultados del caso experimental 2	65
Figura 50. Resultados del caso experimental 3	65
Figura 51. Resultados del caso teórico 1	66
Figura 52. Resultados del caso teórico 2	67
Figura 53. Resultados del caso teórico 3	67
Figura 54. Resultados de la prueba Relé	68

ÍNDICE DE TABLAS

Tabla 1. Combinaciones válidas y sus componentes activos	35
Tabla 2. Ejemplo de factores de ponderación para distintos días.....	44
Tabla 3.Tarifas eléctricas de las bombas	63
Tabla 4.Casos de estudio	64
Tabla 5. Comparativa de las pruebas experimentales.....	65
Tabla 6. Comparativa de las pruebas teóricas	67
Tabla 7. Comparativa entre prueba relé y casos experimentales.....	68
Tabla 8.Coste de los componentes de la maqueta hidráulica	143
Tabla 9.Coste de los componentes sistema de control	143
Tabla 10.Tiempo invertido en el proyecto por cada tarea	143
Tabla 11. Coste horas personal.....	143

MEMORIA

INDICE DE LA MEMORIA

1. INTRODUCCIÓN	15
2. OBJETO.....	15
3. ALCANCE.....	15
4. ANTECEDENTES.....	16
5. ANÁLISIS DE SOLUCIONES ALTERNATIVAS	16
5.1. SELECCIÓN DE DEPÓSITOS.....	16
5.2. SELECCIÓN DE BOMBAS	16
5.3. SELECCIÓN DE CAUDALÍMETROS	17
5.4. SELECCIÓN DE ELECTROVÁLVULAS	17
5.5. SELECCIÓN DE SENSORES DE NIVEL	18
5.6. SELECCIÓN DEL EQUIPO DE CONTROL	18
5.7. SELECCIÓN DE AMPLIFICADORES.....	18
5.8. SELECCIÓN DEL TIPO DE COMUNICACIÓN	18
6. SOLUCIÓN ADOPTADA.....	19
6.1. DESCRIPCIÓN DE LOS ELEMENTOS	19
6.2. CONEXIONADO HIDRÁULICO	24
6.3. CONEXIONADO ELÉCTRICO	26
6.4. DESARROLLO DEL PROYECTO	27
6.4.1. DESARROLLO HIDRÁULICO Y ELÉCTRICO	27
6.4.2. DESARROLLO DE LA COMUNICACIÓN.....	28
6.5. ESTRUCTURA DE CONTROL DEL CAUDAL	29
6.6. DESCRIPCIÓN DEL PROBLEMA DE OPTIMIZACIÓN	33
6.6.1. MODELO MATEMÁTICO DEL PROBLEMA.....	33
6.6.2. REDUCCIÓN DE LA CARGA COMPUTACIONAL DE LA OPTIMIZACIÓN.....	37
6.6.3. RESTRICCIONES DE VOLUMEN ADICIONALES Y LIMITACIÓN DE LAS CONMUTACIONES	40
6.6.4. ESTADO FINAL DEL PROBLEMA DE OPTIMIZACIÓN	42
6.6.5. PREDICCIÓN DEL FLUJO DE SALIDA	43
6.6.6. FLUJO DE SALIDA REAL.....	44
6.7. CÓDIGO DE MATLAB.....	46
6.7.1. PROGRAMA: INICIALIZA.....	47
6.7.2. PROGRAMA: LLAMA AL OPTI.....	48
6.7.3. PROGAMA: OPTIMIZA BOMBEO FUTURO	53
6.7.4. FUNCIÓN: TIMER LAB.....	56
6.8. CÓDIGO DEL PLC.....	57
6.8.1. PROGRAMA: DEMANDA CAUDAL	57
6.8.2. PROGRAMA: CONTROL BOMBAS.....	59
7. ESTUDIO DE VIABILIDAD	62
7.1. VIABILIDAD TÉCNICA.....	62
7.2. VIABILIDAD ECONÓMICA.....	62
8. RESULTADOS.....	63

9. CONCLUSIÓN Y TRABAJOS FUTUROS	69
9.1. CONCLUSIÓN	69
9.2. TRABAJOS FUTUROS	69
10. BIBLIOGRAFÍA	70
11. ANEXOS	71
11.1. LISTADO DEL CÓDIGO DE MATLAB	73
11.1.1.PROGRAMA: INICIALIZA.....	73
11.1.2.PROGRAMA: LLAMA AL OPTI.....	75
11.1.3.PROGRAMA: OPTIMIZA BOMBEO FUTURO	81
11.1.4.FUNCIÓN: TIMER LAB.....	86
11.2. LISTADO DEL CÓDIGO DE CODESYS.....	88
11.2.1.PROGRAMA: DEMANDA CAUDAL	88
11.2.2.PROGRAMA: CONTROL BOMBAS.....	91
11.2.3.VARIABLES GLOBALES.....	93
11.2.4.CONFIGURACIÓN E/S PLC.....	94
11.2.5.CONFIGURACIÓN DE TAREAS	95
11.3. FICHAS TÉCNICAS.....	96
11.3.1.BOMBA SUMERGIBLE.....	96
11.3.2.AMPLIFICADOR.....	97
11.3.3.CAUDALÍMETRO.....	99
11.3.4.ELECTROVÁLVULA.....	100
11.3.5.SENSOR DE NIVEL ENDRESS HAUSER	101
11.3.6.PLC	103
11.3.7.MODULOS WAGO.....	105

1 INTRODUCCIÓN

La optimización es la búsqueda y selección del mejor elemento entre un conjunto según un cierto criterio. En nuestro caso y con un carácter más matemático, optimizar consiste en encontrar un máximo o mínimo de una función objetivo sujeta a unas ciertas restricciones. En este proyecto se busca minimizar el coste operativo de una instalación de suministro de agua y conseguir el mayor ahorro monetario posible.

Por otra parte, en la actualidad la necesidad de ahorrar energía y aumentar la eficiencia en el consumo para combatir el cambio climático y los efectos adversos que numerosos expertos predicen, es de vital importancia. Por ello cada vez más, las industrias, los grandes consumidores y el público en general están más comprometidos con el ahorro energético y buscan medidas para reducir sus consumos.

En el ámbito hidráulico, como en estaciones de bombeo para abastecimiento, es de lógica intentar tener los depósitos de agua llenos al final del periodo de valle para intentar conseguir el mayor ahorro posible. Sin embargo, las decisiones a lo largo del día de cuando bombear y cuando no, vienen determinadas por la demanda de caudal de ese día en un formato de histéresis, en el cual una vez se baja de cierto nivel se llena el depósito para poder seguir ofreciendo servicio.

La finalidad última de este TFM es, por lo tanto, controlar de forma activa el proceso de decisión del llenado de dichos tanques de forma que se consiga el mayor ahorro económico posible, con lo que también se contribuye a aplanar en la medida de lo posible la curva de demanda eléctrica y con ello la eficiencia del sistema eléctrico español.

2 OBJETO

El objeto del presente proyecto es la realización de una maqueta a escala de un sistema hidráulico y la implementación de un control óptimo del sistema de bombeo para que teniendo en cuenta unos periodos tarifarios dados se tenga el menor coste posible.

El proyecto surge de la necesidad de trasladar una simulación únicamente virtual anterior a un sistema físico real, comprobar su funcionamiento y depurar los errores inherentes del uso en la práctica.

3 ALCANCE

La realización de este TFM abarca: realizar el montaje de una maqueta hidráulica, incluyendo su conexionado tanto eléctrico como hidráulico, el control del sistema desde un PLC, la optimización de los periodos de bombeo desde un PC y la comunicación entre la optimización y el sistema físico en tiempo real.

Para la consecución de este proyecto se ha tenido que realizar una selección de los elementos hidráulicos y electrónicos, su posterior conexionado, desarrollo del control PLC de la maqueta, comunicación y control desde el PC a el PLC y experimentación.

Este proyecto no abarca estudios hidráulicos de pérdidas en conducción ni puntuales puesto que se centra en el ámbito del control y está pensado principalmente para comprobar el control de un posible sistema hidráulico real ya calculado, construido y en funcionamiento.

4 ANTECEDENTES

Se han desarrollado trabajos teóricos (Sanchis, 2017) y de simulación por ordenador sobre el tema de este trabajo y este proyecto viene a suplir la necesidad de una puesta en práctica de los conceptos desarrollados en estos.

5 ANÁLISIS DE SOLUCIONES ALTERNATIVAS

5.1 SELECCIÓN DE DEPÓSITOS

Debido al reducido espacio y presupuesto disponibles, la capacidad de los depósitos a seleccionar debe ser reducida. En el laboratorio se dispone de un conjunto de depósitos usados habitualmente para prácticas que cumple con estos requisitos, por lo tanto, con el fin de ahorrar en costes se hará uso de estos. Los depósitos que actuarán como pozos (en adelante pozos) son de mayores dimensiones que los que actuarán como tanques (en adelante tanques) ya que de esta forma se asegura que siempre se tendrá disponibilidad de agua.

5.2 SELECCIÓN DE BOMBAS

En el ámbito industrial lo más común es utilizar bombas trifásicas para trasegar el agua en instalaciones hidráulicas y variadores de frecuencia para la regulación de los caudales que estas entregan. Sin embargo, para este caso esa configuración no es viable al otorgar caudales demasiado grandes, y puesto que se pretende realizar el experimento a un menor tamaño y coste, se ha optado por utilizar bombas de corriente continua sumergibles.

En un principio se intentaron algunas pruebas de funcionamiento con una bomba en serie en la salida inferior de los tanques, tratando de simular una curva de demanda de caudal diaria estándar. Sin embargo, el caudal no podía ser controlado de forma efectiva debido a la presión excesiva que se tenía en el fondo. Por lo tanto, se ha optado por introducir bombas sumergibles también en los tanques, además de en los pozos, y extraer el agua por uno de los orificios superiores de este. De esta manera se tiene un mayor control sobre el caudal de demanda.

Para escalar los caudales y tiempos de la maqueta de forma que sean lo más fieles posible a un sistema real con un tanque de 500 m³ y una bomba de llenado de 200 m³/h, se realizan los siguientes cálculos:

$$\frac{V_{tanqueReal}}{V_{tanqueMaqueta}} = \frac{500000 \text{ litros}}{5 \text{ litros}} = 100000$$

Esto indica que el volumen de los tanques de la maqueta es 100 mil veces más pequeño que el tanque real tomado como modelo y por lo tanto para tener unos tiempos de operación similares a este el caudal de las bombas a seleccionar debería ser de igual manera 100 mil veces menor al real, unos 0.03 litros por minuto. Lo cual es un caudal muy bajo y que nos daría enormes tiempos en el experimento. Por ello también se escala temporalmente. Tomando la escala 1:60 se tiene que, 1 segundo simulado son 60 segundos en la realidad y con ello se consigue el caudal a utilizar sea:

$$Q_{Maqueta} = \frac{Q_{Real}[\text{litros por minuto}]}{\left(\frac{100000}{60}\right)} = \frac{3333.33}{\left(\frac{100000}{60}\right)} = 2 \text{ litros por minuto}$$

Si bien los caudales a utilizar son de entorno a los 2 litros por minuto, los caudales comerciales bajos en bombas sumergibles rondan los 10 o 12 litros por minuto y presentan opciones de modelos entre los 12 y 24 V. Por lo tanto, se eligieron las bombas entre estos caudales, esto además aporta mayores posibilidades de caudal en caso de ser necesarias. Por otra parte, puesto que, para una misma potencia a mayor voltaje menor corriente, se selecciona la opción de 24V. Esta opción es mejor puesto que reduce la corriente total que consume el sistema y que está restringida por la fuente de alimentación a utilizar.

5.3 SELECCIÓN DE CAUDALÍMETROS

Para seleccionar los caudalímetros se ha tenido en cuenta las magnitudes en las que la bomba puede funcionar, puesto que estas están en torno a los 10 l/min el rango de lectura del caudalímetro debería ser entre 0 y 10 l/min o con un máximo inferior, ya que, si bien la selección de las bombas se ha realizado en torno a los 10 l/min nominales, los caudales necesarios son inferiores. La razón de esto es que para realizar una correcta dinámica de llenado de los depósitos a lo largo del día se requieren bajos caudales que no llenen el tanque demasiado rápido.

5.4 SELECCIÓN DE ELECTROVÁLVULAS

Debido a la necesidad de controlar los flujos de caudal y probar distintas configuraciones de conexiones hidráulicas se hace uso de diversas electroválvulas de 2 vías. Las presiones que deberán soportar no son muy elevadas por lo que se pueden despreciar, y no suponen una restricción. Por ello se seleccionarán electroválvulas de riego de tamaño similar al de los otros componentes del sistema, entorno a los 20 mm de sección de paso.

5.5 SELECCIÓN DE SENSORES DE NIVEL

En el laboratorio ya se dispone de un sensor de nivel capacitivo funcional por lo que solo se requiere la adquisición de un segundo sensor de nivel para el otro tanque. El criterio principal para su selección es que el rango de medición del sensor este alrededor de 50 cm de longitud. Entre las diversas tecnologías existentes para la medición de nivel tal como ultrasonidos o por radar, se ha optado por la tecnología capacitiva al adaptarse mejor al presupuesto disponible y ofrecer una precisión más que satisfactoria para la tarea que se le asigna.

5.6 SELECCIÓN DEL EQUIPO DE CONTROL

Para la gestión de los diferentes componentes del sistema se precisa de un equipo de control. Para esta tarea existen diversos sistemas que podrían desempeñar tal función tales como un sistema Arduino, un PLC o un PC. Sin embargo, en este proyecto nos hemos decantado por la opción de utilizar un PLC puesto que es un equipo cuyo uso está muy extendido en la industria y ofrece multitud de posibilidades para el control.

5.7 SELECCIÓN DE AMPLIFICADORES

El control de los actuadores y sensores se realizará con un PLC, pero este solo responde a una señal en tensión o en corriente, es decir, no entrega potencia. Por lo tanto, se requerirá del uso de amplificadores para traducir una señal en tensión a una potencia determinada que haga funcionar a las bombas en diferentes regímenes.

Estos amplificadores requerirán poder ser alimentados como mínimo con 24 V y 1 A ya que son los valores nominales entre las bombas estudiadas para este proyecto. También es necesario que tengan una entrada analógica en tensión entre 0-5 V o 0-10V, de forma que un módulo de salida en tensión del PLC pueda estar conectado a esta y controlar la tensión que recibe la bomba.

5.8 SELECCIÓN DEL TIPO DE COMUNICACIÓN

Tras elegir realizar el control del sistema físico vía PLC, queda elegir el tipo de comunicación entre el PLC y el PC donde se realizará la optimización. Las opciones que se barajan son el uso de un servidor OPC y la comunicación directa vía Modbus TCP/IP. Siendo la primera una mejor opción para el manejo y la integración de la información desde distintos dispositivos, softwares y bases de datos; mientras que la segunda opción permite una mayor velocidad de intercambio de información y no hay necesidad de adquirir una licencia.

6 SOLUCIÓN ADOPTADA

La maqueta consta de 4 depósitos (2 tanques y 2 pozos), 4 bombas para el llenado y vaciado de los tanques, 4 caudalímetros, 4 electroválvulas y 2 sensores de nivel. Todo ello controlado desde un PLC que recopila la información de los sensores y envía las señales a los actuadores. A su vez las acciones del PLC estarán supeditadas al resultado de una optimización periódica realizada en el PC desde el programa Matlab, el cual finalmente se comunicará vía Modbus con el PLC.

6.1 DESCRIPCIÓN DE LOS ELEMENTOS

A continuación, se presenta las características principales de los equipos finalmente seleccionados:

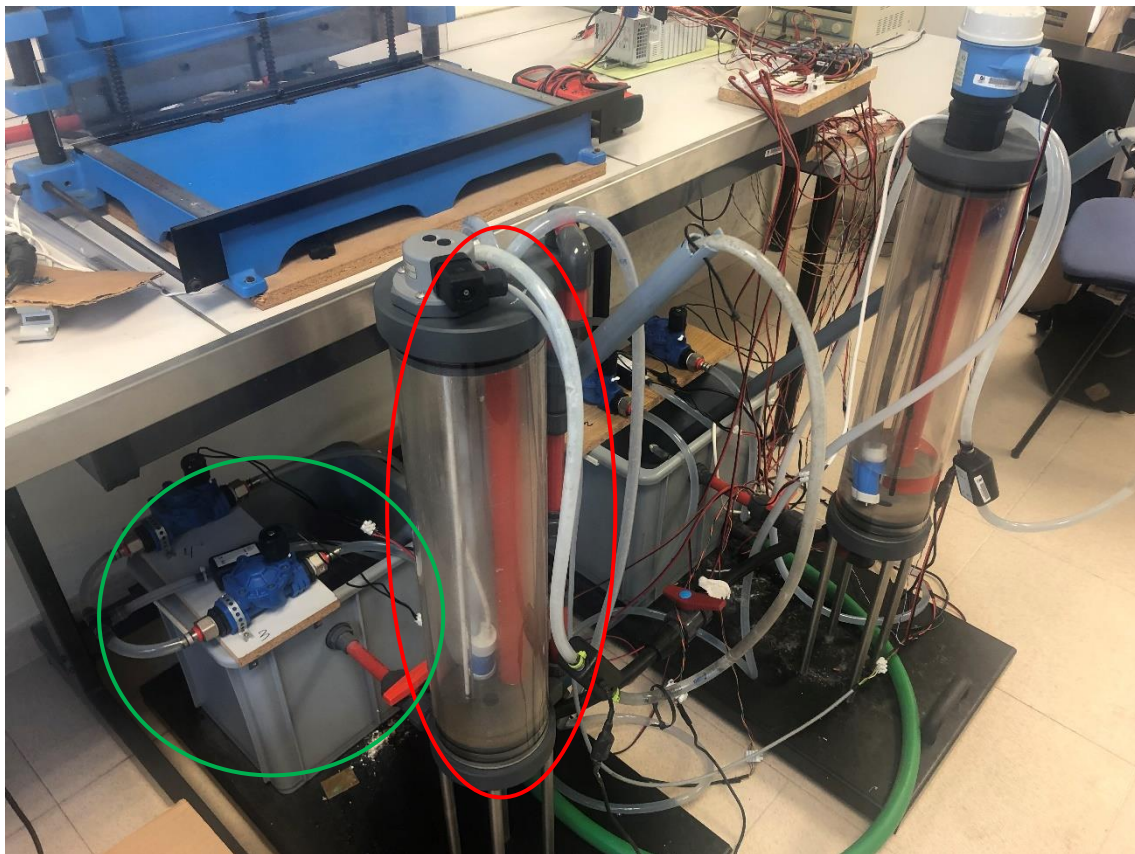


Figura 1. Pozos (verde) y Tanques (rojo) de la maqueta

Depósitos:

Pozos

- Cantidad: 2
- Capacidad: 20 litros cada uno
- Forma: Rectangular

Tanques

- Cantidad: 2 unidades
- Capacidad: 5 litros cada uno
- Forma: cilíndrica



Figura 2. Bomba sumergible 24V

Bombas:

- Tipo: Sumergible
- Marca: COMET
- Modelo: ELEGANT 24V
- Cantidad: 4 unidades
- Caudal nominal: 10 l/min
- Alimentación: 24VDC



Figura 3. Caudalímetro

Caudalímetros:

- Tipo: Ultrasónico

- Marca: CYNERGY3
- Modelo: UF08B100
- Cantidad: 4 unidades
- Caudal máximo: 8 l/min
- Caudal mínimo: 0.1 l/min
- Alimentación: 24 VDC
- Señal de salida: 0-5Vdc o 4-20mA
- Periodo de muestreo: 250 ms

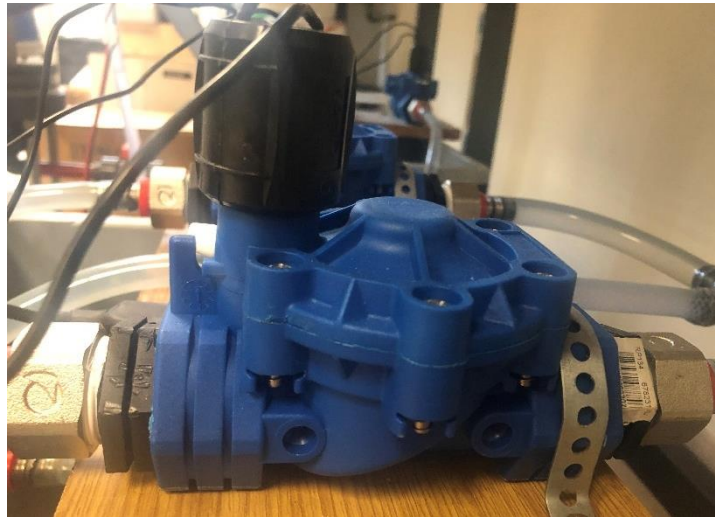


Figura 4. Electroválvula

Electroválvulas:

- Tipo: 2 vías, Normalmente Cerrada
- Marca: Orbit
- Modelo: 57100
- Cantidad: 4 unidades
- Alimentación: 24V



Figura 5. Sensores de nivel capacitivos

Sensores de nivel:

- Tipo: Capacitivo

- Marca: Elobau/ Endress Hauser
- Modelos: 212KFA02/ Liquicap T FMI21
- Cantidad: 2 unidades
- Rango: 500 mm
- Alimentación: 24VDC
- Señal de salida: 0-10Vdc/ 4-20mA

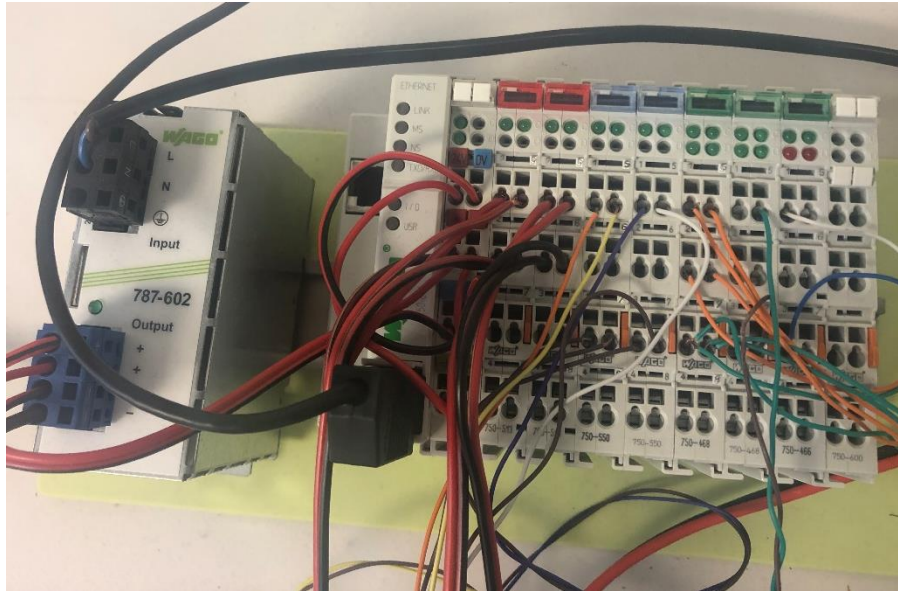


Figura 6. PLC Wago y periféricos entrada-salida

Equipo de control:

- Tipo: PLC
- Marca: Wago
- Modelo: 750-841
- Alimentación: 24V

Módulos

- Modelo: 750-468
- Cantidad: 2 unidades
- Tipo: Entrada analógica
- Número de entradas: 4
- Voltaje de entrada: 0-10Vdc

- Modelo: 750-550
- Cantidad: 2 unidades
- Tipo: Salida analógica
- Número de salidas: 2
- Voltaje de salida: 0-10Vdc

- Modelo: 750-513
- Cantidad: 2 unidades
- Tipo: Salida relé
- Número de salidas: 2

- Máximo voltaje de conmutación: 30Vdc
- Modelo: 750-466
- Cantidad: 1 unidades
- Tipo: Entrada analógica
- Número de entradas: 2
- Corriente de entrada: 4-20 mA
- Modelo: 750-600
- Tipo: Modulo Final
- Cantidad: 1 unidad



Figura 7. Amplificador de potencia

Amplificadores:

- Marca: United Automation
- Modelo: DMC X10335
- Cantidad: 4 unidades
- Tensión continua máxima: 24Vdc
- Corriente máxima: 40A
- Rango tensión de entrada: 0-5Vdc



Figura 8. Fuente de alimentación

Fuente de alimentación:

- Marca: GW INSTEK
- Modelo: GPC-3030D
- Cantidad: 1 unidad
- Potencia: 195W
- Tensión de alimentación: 90-253 Vac
- Intensidad máxima: 3 A en paralelo, 6 A en serie
- Número de salidas: 3

Softwares utilizados:

- Matlab 2018
- Codesys V2

6.2 CONEXIONADO HIDRÁULICO

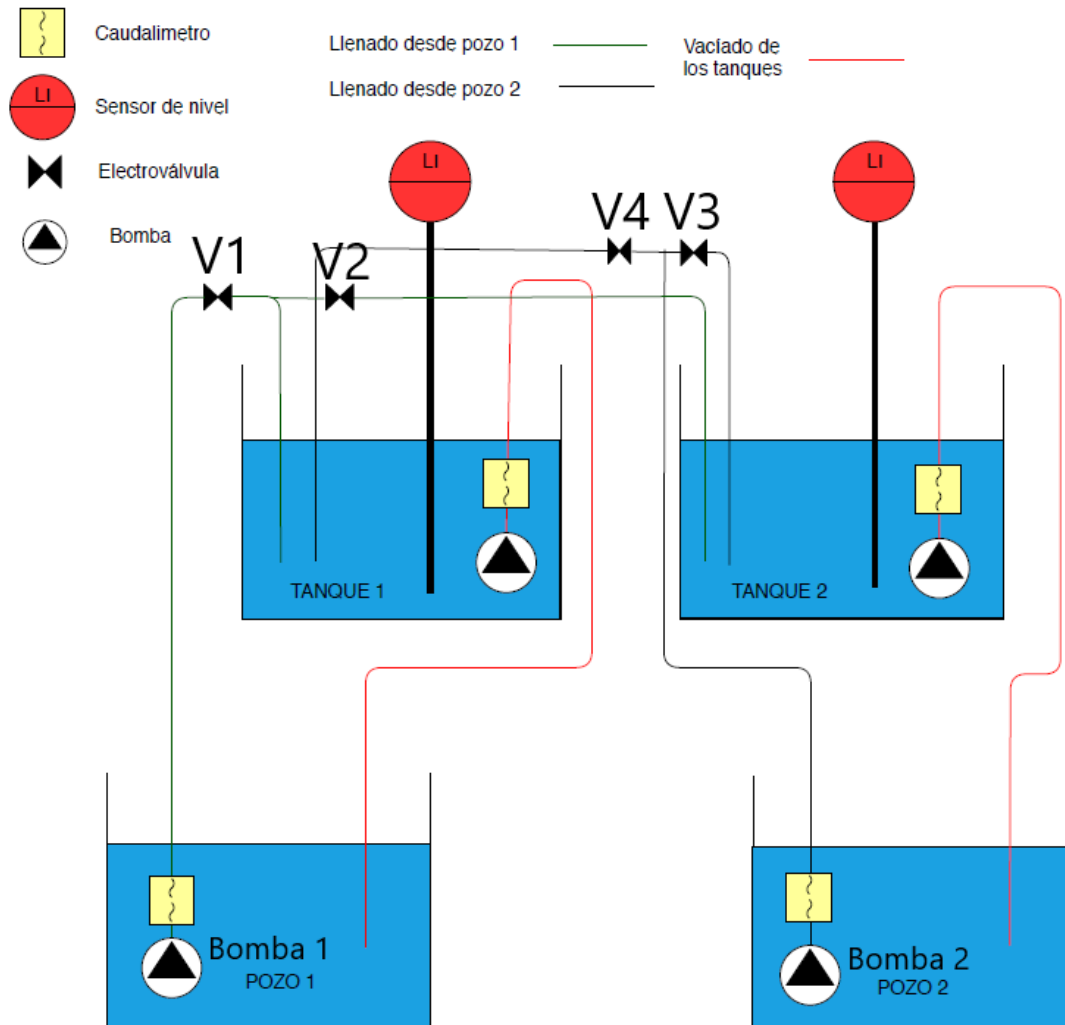


Figura 9. Esquema hidráulico de la maqueta

En la Figura 9. **Esquema hidráulico de la maqueta** podemos observar un esquema del conexionado hidráulico de la maqueta a realizar; 4 depósitos, 2 tanques y 2 pozos, cada uno de estos con su correspondiente bomba y caudalímetro. Finalmente 2 sensores de nivel cada uno en uno de los tanques y electroválvulas que controlan que el bombeo desde cada pozo se pueda dirigir a uno u otro tanque.

Para el montaje del sistema de llenado se colocará una bomba en el fondo del pozo conectada a un caudalímetro. Posteriormente, pasando a través de una desviación en forma de T donde cada ramal conducirá a uno de los tanques, se atravesará una electroválvula que permitirá controlar si se llena o no el correspondiente tanque. Se hará de forma análoga con el otro pozo y tanque. Cada uno de estos elementos estará unido entre si mediante un tubo de plástico PVC y se usarán arandelas de plástico en el interior de estos y bridas en el exterior para sellar las uniones y evitar la fuga del agua.

En el caso del sistema de vaciado de los tanques, las bombas se introducirán por una abertura superior y se colocarán en el fondo de estos. Las bombas de vaciado al igual que las de llenado se conectan a un caudalímetro y posteriormente desembocan en un tubo rígido de mayor diámetro conectado a su pozo más cercano.

Finalmente, la medición del nivel de los tanques se realizará mediante un sensor de nivel capacitivo de varilla. Dicha varilla se introducirá por otra de las aberturas superiores de los tanques. Las medidas de los sensores no se verán perturbadas por el llenado ya que este último se realiza a través de un tubo rígido de plástico que llega hasta prácticamente el fondo del tanque y protege de esta forma al sensor de la mayor parte de las fluctuaciones del llenado.

6.3 CONEXIONADO ELÉCTRICO

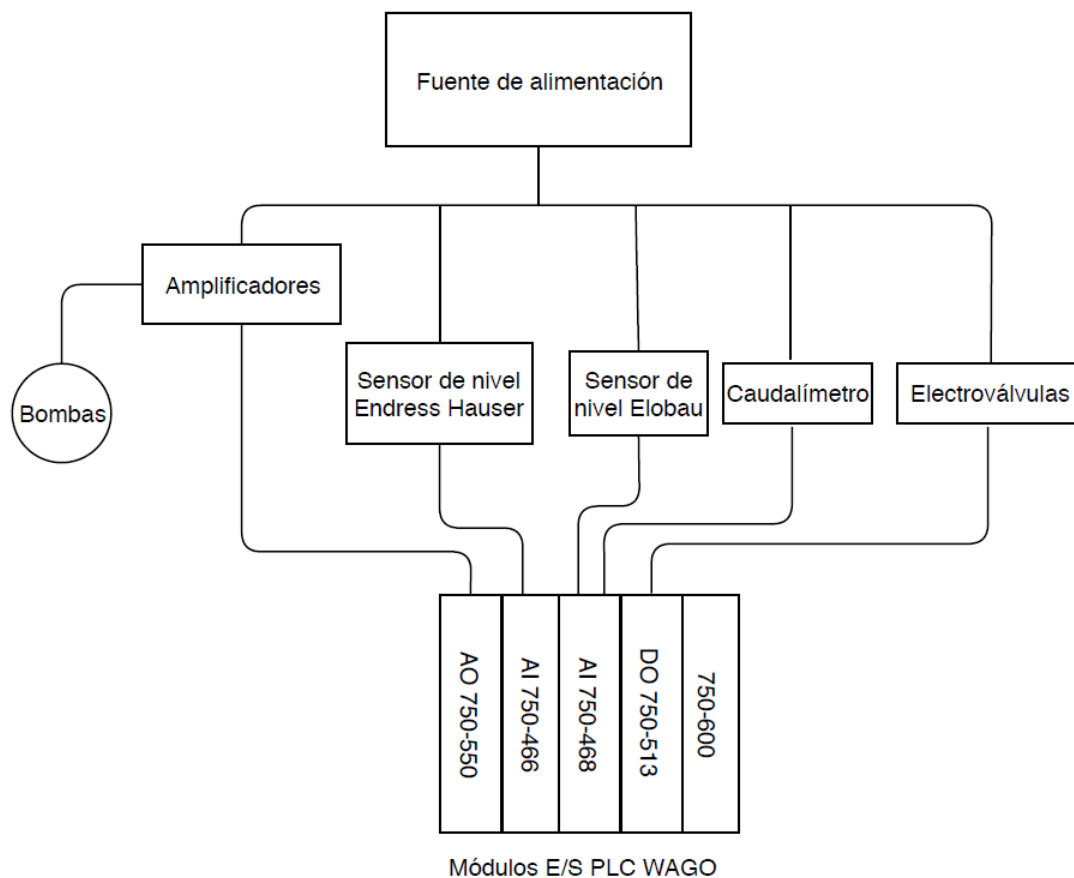


Figura 10. Esquema eléctrico general

En la Figura 10. **Esquema eléctrico general** se puede observar un esquema básico de las conexiones eléctricas entre los componentes del sistema. En la parte superior se sitúa la fuente de alimentación que suministra potencia al resto de componentes y en la parte inferior podemos observar los distintos módulos E/S del PLC a los que van conectados. Los amplificadores están conectados a una salida analógica en tensión (750-550), con rango entre 0 y 10V, que aporta la señal necesaria para controlar la amplificación de potencia y con ello las respectivas bombas. Por otra parte, existen dos sensores de nivel, uno que se conecta a una entrada analógica en tensión (750-468) con rango de 0 a 10V y otro a una entrada analógica en intensidad (750-466) con rango de 4 a 20mA. Por último, el caudalímetro envía señales de datos a una entrada analógica de tensión (750-468) y las electroválvulas se controlan mediante una salida relé (750-513).

Las diferentes conexiones eléctricas entre cada componente, su alimentación y el sistema de control PLC se detallarán en los planos.

6.4 DESARROLLO DEL PROYECTO

6.4.1 DESARROLLO HIDRÁULICO Y ELÉCTRICO

Una vez realizado el montaje de la planta piloto se pasa a realizar las comprobaciones de los distintos elementos. Una comprobación simple pasa por poner en marcha las bombas y activar las correspondientes electroválvulas de forma que llenemos ambos depósitos sin reparar en la configuración concreta a usar, tanto se puede llenar el tanque 1 con el agua del pozo 1 o con el del pozo 2 e igual en el caso del tanque 2. De esta forma se comprueba que ambas bombas funcionan y responden correctamente a un incremento o descenso del voltaje aplicado desde el PLC. La señal del caudalímetro cambia también acorde a los valores dados a su respectiva bomba y la señal de los sensores de nivel cambia conforme los tanques se llenan o vacían.

Un parámetro a determinar es el rango de caudal en el que funcionaban las bombas, por ello en primer lugar se explicarán las relaciones entre las unidades del PLC y las unidades de las señales de medida. Las salidas y entradas analógicas de los distintos módulos del PLC envían y reciben señales en voltios o miliamperios, sin embargo, internamente estos valores se convierten linealmente en unidades digitales del PLC. Los datos de E/S se almacenan en variables WORD las cuales abarcan un rango de valores enteros entre 0 y 32767, por lo tanto, para el tipo de salida analógica de las bombas con un rango de tensiones entre 0 y 10V, 10V se correspondería con el máximo valor de las unidades digitales tipo WORD, 32767, y 0V con 0 unidades digitales. De igual forma funciona el tratamiento del resto de señales de entrada y salida. Por ejemplo, en una entrada digital con rango 0-10V, un valor de 5V se traduciría en un valor digital de 16384.

Para la configuración de las mediciones de caudal se tiene en cuenta el rango de medición de los caudalímetros, el cual es de 0 a 8 litros por minuto. También se tiene en cuenta la señal de datos que envía el caudalímetro que tiene un rango entre 0 y 5V. Sabiendo esto se establece una conversión en base a una regla de tres entre el rango de caudales del sensor y el voltaje de su señal de datos. Con ello y utilizando la conversión voltaje-unidades digitales explicada anteriormente se puede conocer el caudal que trasiegan las bombas. El mismo método es usado para las mediciones de nivel de los depósitos.

Para conocer el rango de caudales en el que trasegaban las bombas en su estado inicial se les alimento en un momento a 24 V y después a el voltaje mínimo en el que se conseguía trasegar el agua, alrededor de 2 V, mientras se median los caudales obtenidos. En un primer momento el rango de estos caudales estaba demasiado desplazado hacia valores altos, entre 2 l/min y 6 l/min. Por lo tanto, y teniendo en cuenta que la capacidad de los depósitos es de unos 5 litros cada uno, se buscó que al menos el límite inferior estuviera por debajo de 1 litro por minuto y el límite superior alrededor de 2 litros por minuto.

Para conseguir que el rango de caudal de las bombas se desplace hacia valores más bajos la solución pasa por aumentar la resistencia hidráulica total del sistema. Primeramente, se trató de implementar una medida rápida y sencilla como es doblar los tubos de los ramales de extracción y llenado, de tal forma que al pasar por un estrechamiento la resistencia hidráulica aumentase y bajase el caudal que se puede

bombear. Sin embargo, esta configuración presentaba un comportamiento muy escalonado y no permitía ajustar con un mínimo de precisión el caudal mínimo. Por otra parte, otra de las consignas que se buscaba en la descarga del agua demandada, era que el límite superior de caudal fuera como mínimo 4 veces el límite inferior para así poder tener una curva de demanda lo más parecida a una real a escala. Este primer arreglo no lo conseguía. Además, con el paso del tiempo el grado de doblez del codo realizado en el tubo cambiaba y ello hubiese afectado a los resultados obtenidos en los experimentos.

Por todo ello se decide buscar una alternativa más estable, la cual es obstruir la salida de los distintos tubos con un tapón cilíndrico de madera y perforarlo con un taladro. Se han ido probando distintos diámetros de taladro y comprobando los límites superior e inferior que se pueden obtener hasta alcanzar los deseados, siendo estos en torno a 0.6 y 2 l/min con un agujero de 3 mm de diámetro en todas las salidas. Con este último diámetro se consigue poder bombear en un rango de caudal lo suficientemente bajo de forma estable y se tiene una relación entre el límite superior e inferior cercana a 4. Si esta configuración no hubiese resultado estable, se podría plantear realizar un orificio en la sección del tubo que se encuentra en el interior del tanque para que una parte del caudal volviese a este. De esta forma se conseguiría que la bomba trabajase en tensiones más altas, a la vez que se consiguen caudales de extracción bajos.

Habiendo conseguido que las bombas trabajasen a caudales más bajos surgieron problemas que se presentaban al encender las bombas. El cambio de estar paradas a ponerse a bombear era bastante inestable y se formaban oscilaciones con relativamente altos tiempos de establecimiento. Este dilema no afectaría a las bombas de extracción de agua ya que simulan el caudal de demanda de una ciudad y por lo tanto están encendidas en todo momento durante los experimentos. Sin embargo, es otro caso con las de llenado, para solucionar este problema lo que se hizo fue establecer una saturación mínima del controlador PI tal que la tensión de control de las bombas estuviese entorno a los 1,83V. Este valor corresponde al punto de equilibrio en el que el agua se mantiene en el punto más álgido de la conducción sin llegar a verter en el tanque. De esta forma la bomba en su estado de trabajo mínimo no está parada, sino que preparada para comenzar a trasegar agua y el transitorio de arranque es mucho más corto y suave. Otra opción que se planteó para solucionar este transitorio inicial fue inicializar un arranque al valor que se desee durante un tiempo determinado y después dejar actuar al PID, sin embargo, esta se descartó ya que el tiempo que va a estar encendida la bomba y su frecuencia de encendido es variable y dependiente del resultado de las optimizaciones, por lo tanto, este método es poco práctico e inviable.

6.4.2 DESARROLLO DE LA COMUNICACIÓN

A continuación, se decide testear la correcta comunicación entre el PC y el PLC vía MODBUS. Esta comprobación se ha realizado con un experimento simple en el que se llenan los tanques con el agua de sus respectivos pozos y al llegar a una altura determinada se para el llenado y se activan las bombas de vaciado hasta alcanzar un mínimo. Es decir, se implanta un control todo nada. El control de los actuadores y sensores se realizará desde el PC gracias a la comunicación MODBUS. De esta forma se comprueba que el PC

recibe y manda todas las señales correctamente y que este intercambio de datos se realiza con una rapidez suficiente.

6.5 ESTRUCTURA DE CONTROL DEL CAUDAL

Para obtener una buena gestión del sistema de llenado y vaciado de los tanques, se ha implementado un bucle de control sobre cada bomba de agua. De esta forma se controla en cada momento el agua que se trasiega con una mayor precisión y se consiguen unos mejores resultados en los experimentos.

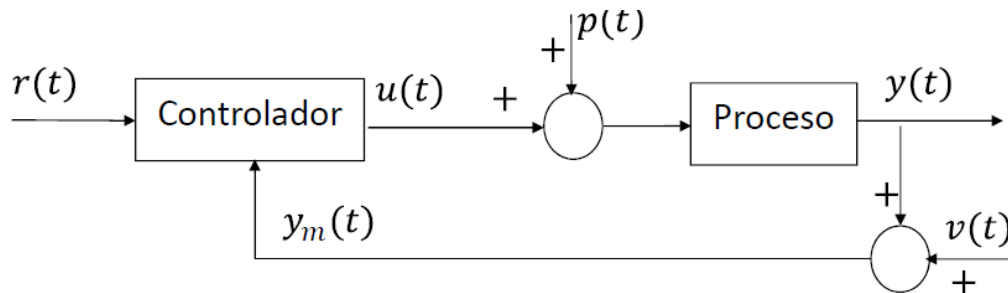


Figura 11. Bucle de control en lazo cerrado

Entre las distintas estrategias de control que se podrían implementar se decidió establecer un bucle de control realimentado (Figura 11) para el sistema de llenado ya que es una estructura sencilla que no encarece el proyecto y permite utilizar un equipo relativamente impreciso y económico manteniendo un buen grado de exactitud. El control también se ha implementado en la extracción, aunque esta no forma parte de los objetivos de control puesto que interesa que exista una cierta fluctuación y aleatoriedad en el comportamiento del vaciado para hacer el experimento más realista.

En primer lugar, se ha hecho la labor de identificar el sistema, dando una entrada escalón a las distintas bombas y guardando tanto esa entrada como los datos registrados por el caudalímetro. Con estos datos y la ayuda de las herramientas Java desarrolladas por Roberto Sanchis se ha obtenido el modelo del proceso y el controlador a implementar.

De entre dichas herramientas utilizaremos únicamente 2, que son:

- Time response based system Identification. Aplicación Java a la que se le cargan los ficheros de datos experimentales en formato ascii y que permite la identificación experimental del proceso y guardar el modelo en un fichero.
- Model based PID design. Esta aplicación Java permite leer el modelo identificado anteriormente, diseñar y simular un PID mediante la respuesta en frecuencia.

A continuación, se expondrán los modelos tanto del proceso como del controlador y su método de obtención a través de las herramientas Java.



Figura 12. Carga de datos - Herramienta Java de identificación

En primer lugar, se cargan los datos de entrada, salida y temporales, y se obtiene el modelo del sistema físico. En nuestro caso hay que marcar la casilla, que se observa en la Figura 12. **Carga de datos - Herramienta Java de identificación**, de “Tiempo en ms” ya que el registro de datos en Codesys los entrega con el vector temporal en milisegundos.

El caudalímetro registra una medida cada 250 ms, por lo tanto, existe una restricción en cuanto al tiempo de muestreo disponible y como se puede observar en la Figura 13, los datos se presentan con un formato escalonado. Estos 250 ms son un tiempo grande en comparación con la dinámica del caudal de las bombas por lo que se eligió utilizar la opción de trabajar en digital durante el diseño del PID.

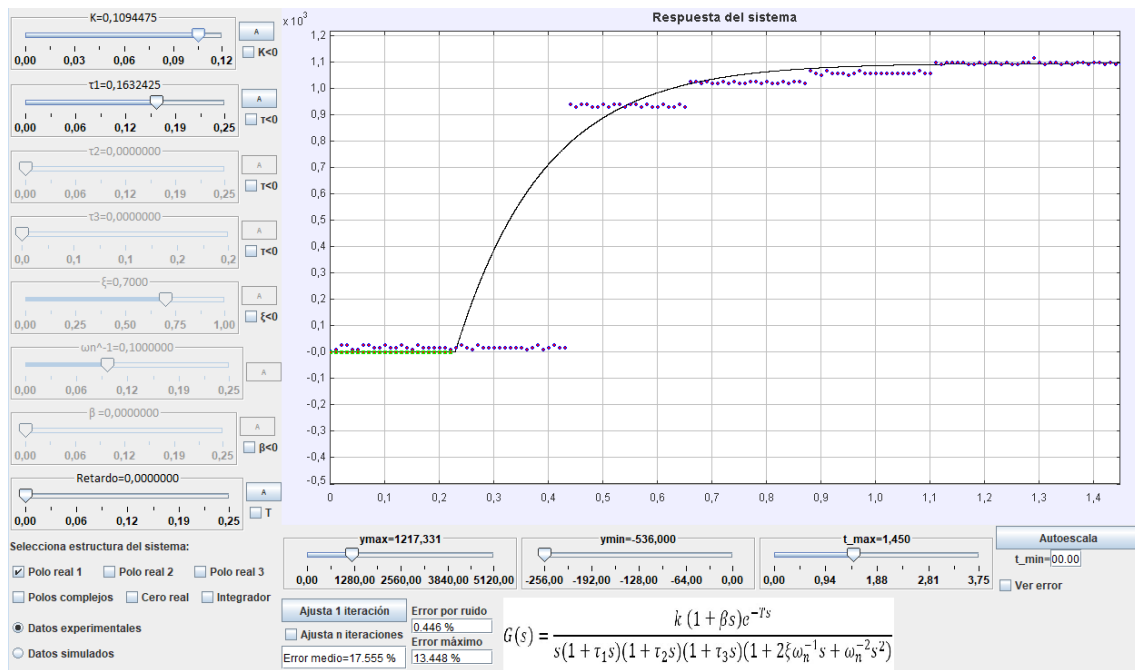


Figura 13. Identificación- Herramienta Java de identificación

Durante la identificación se decidió ajustar los datos a un modelo de primer orden ya que no se dispone de mayor información para afinar más el modelo. Se establece por lo tanto una ganancia (K) y un polo (Tau) de forma que coincida lo máximo posible con la salida y por último se guarda el modelo en el disco.

Para ajustar el modelo primeramente se establece la ganancia moviendo la deslizador correspondiente hasta que la altura final de la línea de modelado coincida con el valor final de los datos y a continuación se implementa el valor del polo. Este se mueve de forma que se hace coincidir la curva con la parte izquierda de los diferentes escalones de la salida puesto que es el momento en el que se registran los datos, y el resto del escalón es una prolongación de esa medida hasta que se obtiene la siguiente muestra.

A continuación, se presenta el modelo del sistema identificado:

$$G(s) = \frac{k}{1 + \tau \cdot s} = \frac{0.109}{1 + 0.163 \cdot s}$$

Tras obtener el modelo del sistema se pasa a diseñar el controlador PID. En este se busca que no sea demasiado agresivo para que no produzca fluctuaciones muy grandes al estar trabajando en torno al límite inferior del rango de caudales de las bombas, pero lo suficientemente rápido como para seguir las referencias. Para comenzar este proceso se abre la aplicación Java de diseño del PID y se carga el modelo obtenido anteriormente. Se establece el periodo de 250 ms y se marca la casilla de hacer el cálculo en digital.

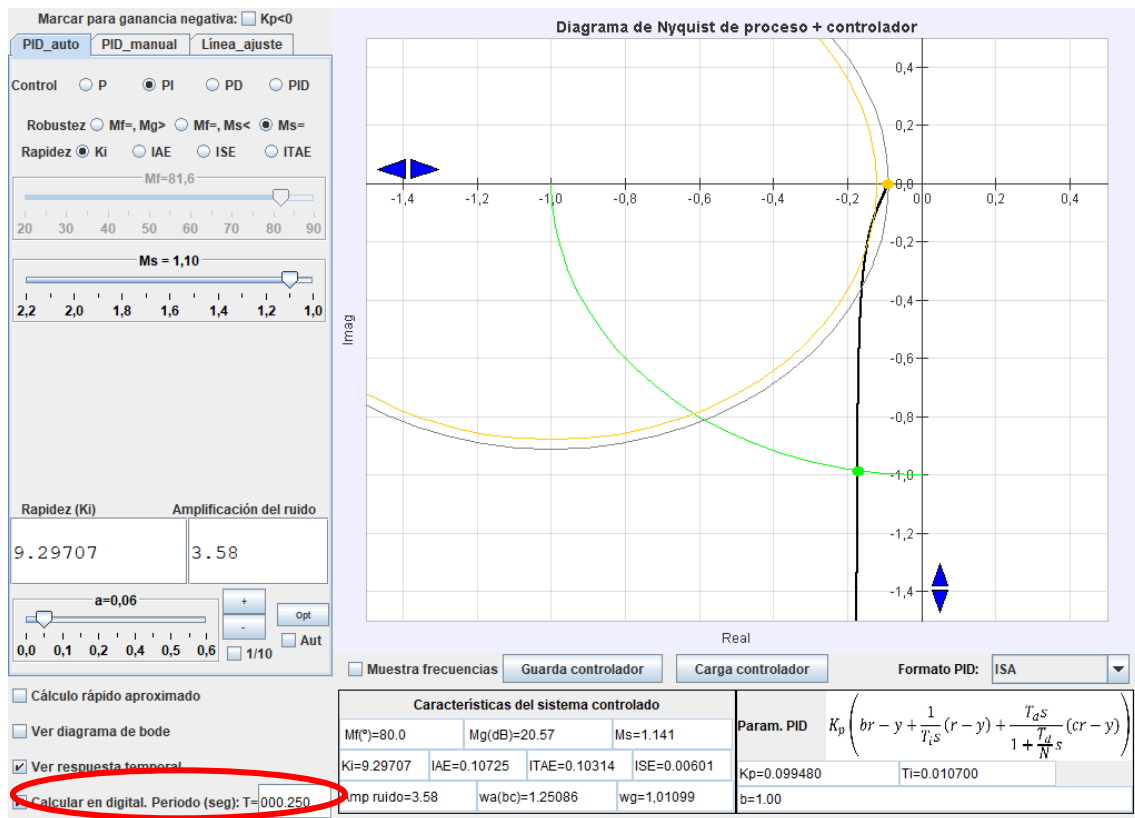


Figura 14. Diseño del PID- Herramienta Java de diseño PID

Los siguientes pasos a realizar son definir un margen de fase y seleccionar la opción PI al tratarse de un modelo de primer orden. El programa calcula el PI más rápido que cumple con el margen de fase establecido. Sin embargo, la respuesta que se obtiene de esa forma es muy rápida y la ganancia extremadamente alta. Esto se debe a que un sistema de primer orden se puede hacer prácticamente infinitamente rápido con un controlador PI. Por ello se ha de hacer más lento reduciendo el parámetro “a” hasta conseguir una amplificación del ruido razonable. Al bajar el valor de esta variable respecto del óptimo se obtienen diseños más lentos y robustos.

Otro método de diseño del PID es el que se observa en la Figura 14 en el que se establece el valor de “Ms” deseado y se ajusta la velocidad del controlador con el parámetro “a” tal y como en el método anterior.

La variable “Ms” es la inversa del radio del círculo, con centro en el punto -1, que hace que este sea tangente a la curva negra, es decir, al diagrama de Nyquist del proceso con el controlador. Es, por lo tanto, la inversa de la distancia mínima entre la curva negra al punto -1. Siendo que, en un Diagrama de Nyquist, cuanto más alejado del punto -1 se encuentre la curva negra más robusto es nuestro diseño, se tiene que a menor valor del parámetro “Ms” mayor exigencia de robustez se le pide al controlador.

Normalmente se suele utilizar valores de “Ms” entorno a los 1.4, sin embargo, en este caso se ha elegido un valor de 1.1 ya que el actuador del que se dispone (las bombas) se hace fácilmente inestable en rangos de caudal cercanos a su límite inferior. Además, se tiene un cierto ruido de medida y se intenta procurar amplificarlo lo menos posible, para que el comportamiento de las bombas no fluctué en exceso.

Cuanto más rápido es el controlador más pequeño es el IAE y a su vez cuanto más pequeño es el IAE de un controlador diseñado para una perturbación escalón, mejor sigue este una referencia variable como es la de las bombas de vaciado. Por ello se ha intentado determinar un equilibrio entre la rapidez y la robustez del controlador, dando una prioridad a la robustez debido a la naturaleza oscilatoria de las bombas a bajos caudales y a que al menos en los bucles de vaciado no se quiere un seguimiento muy preciso puesto que de esta forma se ayuda a simular una curva de demanda con cierto grado de aleatoriedad.

Se han probado distintos valores de K_i tales como 25 o 10, siendo valores más altos de esta relación controladores más rápidos e inestables y valores más bajos, lentos y robustos. Algunos de los parámetros de estos controladores fueron:

$$C(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s}\right) = 0.5 \cdot \left(1 + \frac{1}{0.02 \cdot s}\right) \quad K_i = 25$$

$$C(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s}\right) = 0.5 \cdot \left(1 + \frac{1}{0.08 \cdot s}\right) \quad K_i = 6.25$$

Sin embargo, estos resultaron o bien demasiado inestables o bien lentos. Finalmente se decidió por su valor actual:

$$C(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s}\right) = 0.1 \cdot \left(1 + \frac{1}{0.01 \cdot s}\right) \quad K_i = 10$$

El cual presenta un comportamiento bastante estable y rápido.

6.6 DESCRIPCIÓN DEL PROBLEMA DE OPTIMIZACIÓN

Como ya se ha comentado previamente este proyecto versa entorno a la optimización de las operaciones de un sistema de suministro de agua potable, que consiste en dos tanques llenados por dos pozos de agua. El resultado de la optimización controla qué válvulas y qué bombas se encuentran activas en cada momento teniendo en cuenta los diferentes periodos tarifarios de electricidad y procurando por lo tanto que el coste operativo de la instalación sea el mínimo posible.

El código de optimización desarrollado utiliza el solver estándar “CBC” para resolver el problema, aunque está configurado para poder utilizar también “Mosek” y “glpk” de igual forma. A continuación, se presenta el modelo matemático del problema que establece el índice de coste y las restricciones de la optimización.

6.6.1 MODELO MATEMÁTICO DEL PROBLEMA

La principal restricción de la optimización es que se debe de poder suministrar desde cada tanque el caudal de agua demandado diariamente. Aunque esta demanda de caudal cambia en el tiempo, se puede predecir ya que sigue de forma aproximada un ciclo diario de repetición. Es por ello que la curva de caudal se simula a partir de una serie de Fourier ajustada en base a una curva de un sistema de bombeo de agua real.

Las válvulas se usan para reconfigurar el recorrido hidráulico del sistema haciendo que se pueda llenar un tanque desde distintos pozos. Eso hace que si se tiene un número de bombas (N_p) y un cierto número de válvulas (N_v), el número de combinaciones posibles sea de $2^{N_p+N_v}$. Sin embargo, no todas estas combinaciones son posibles en un sistema real, aunque sí lo sea en la maqueta, y por ello y para reducir la carga computacional de la optimización se decidió restringir estas posibilidades de forma arbitraria. Para definir el número de combinaciones posibles (N_c) se hace uso de una matriz binaria de tamaño $N_c \times (N_p + N_v)$.

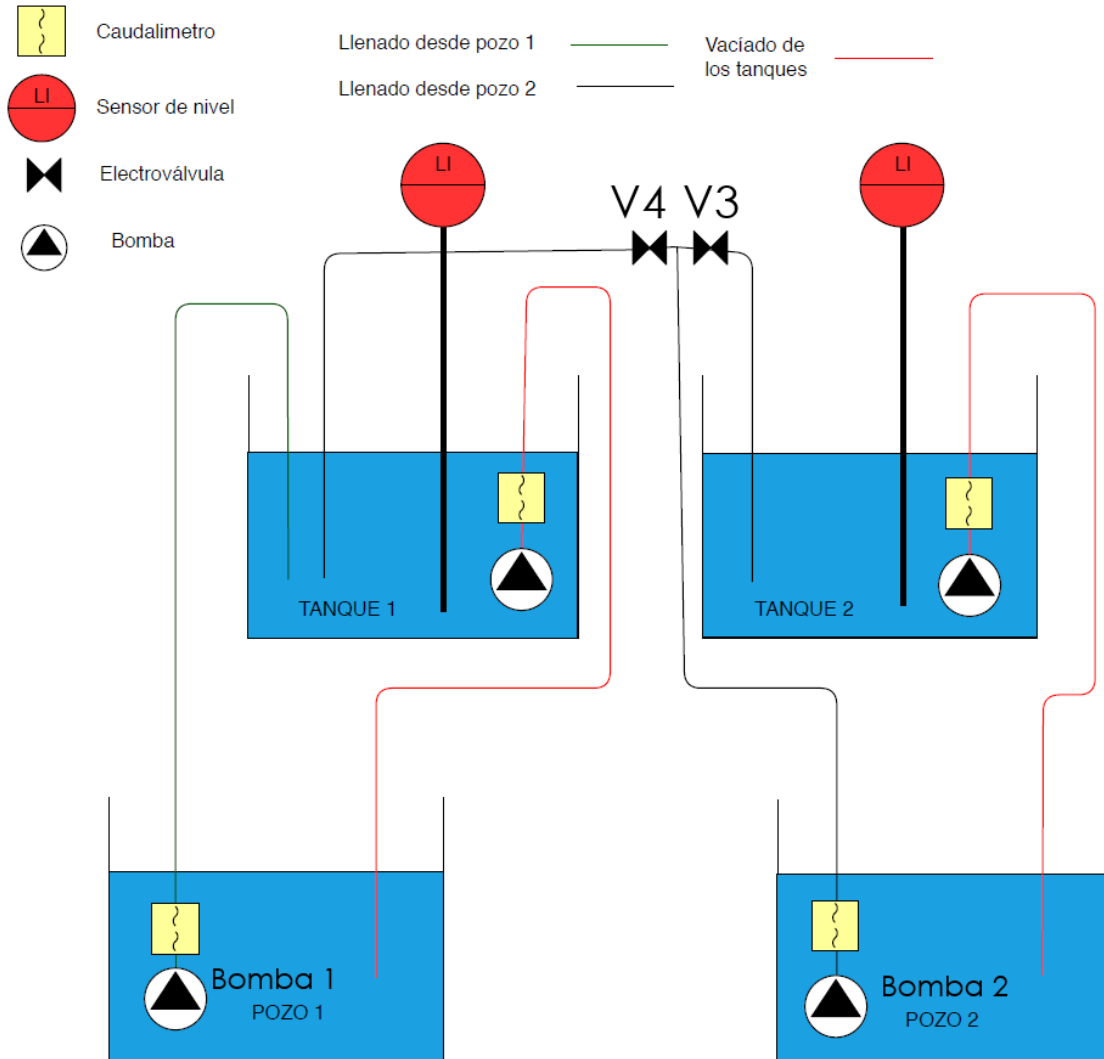


Figura 15. Configuración hidráulica final

Tal y como se puede observar en la Figura 15 la configuración que se ha elegido para simular consta de 2 bombas ($N_p = 2$) y 2 válvulas ($N_v = 2$). Ambas bombas de llenado pueden funcionar a la vez, se puede llenar el tanque 1 o el 2 con la bomba del pozo 2, pero no desde la del pozo 1 puesto que esta solo llena el tanque 1. Teniendo en cuenta esta configuración y sus limitaciones esto da un número total de 6 combinaciones posibles ($N_c = 6$) que se ven reflejadas en la Tabla 1.

Comb	V ₄	V ₃	P ₁	P ₂
0	X	X	0	0
1	X	X	1	0
2	1	0	0	1
3	0	1	0	1
4	0	1	1	1
5	1	0	1	1

Tabla 1. Combinaciones válidas y sus componentes activos

En la Tabla 1 las filas representan cada una de las combinaciones posibles y en las columnas el estado de encendido/abierto o no de las válvulas y bombas correspondientes. La X representa un estado de ambigüedad en el que no importa si las válvulas están abiertas o cerradas ya que no se bombea agua desde el pozo 2 que es el que se ha establecido que dispone de válvulas.

También se modela en matrices el flujo de agua procedente de cada bomba de llenado y el caudal que entra en cada tanque en cada una de las combinaciones. El flujo de caudales desde las bombas de llenado se representa en la matriz F_P , la cual tiene tantas columnas como número de combinaciones y tantas filas como número de bombas de llenado. Por otro lado, el flujo de agua entrante en los tanques se representa en la matriz F_T , la cual tiene de igual forma tantas columnas como combinaciones posibles y tantas filas como tanques. Estos flujos se encuentran medidos en litros por minuto.

$$F_P = \begin{bmatrix} 0 & 2 & 0 & 0 & 2 & 2 \\ 0 & 0 & 1.7 & 1.9 & 1.9 & 1.7 \end{bmatrix} \quad (1)$$

$$F_T = \begin{bmatrix} 0 & 2 & 1.7 & 0 & 2 & 3.7 \\ 0 & 0 & 0 & 1.9 & 1.9 & 0 \end{bmatrix} \quad (2)$$

De igual forma se modela la potencia que consume cada bomba en cada una de las combinaciones posibles. Estas potencias se miden en KW.

$$P = \begin{bmatrix} 0 & 15 & 0 & 0 & 15 & 15 \\ 0 & 0 & 20 & 10 & 10 & 20 \end{bmatrix} \quad (3)$$

El resultado de la optimización debe elegir una entre las 6 posibles combinaciones para aplicarla al estado de las bombas y válvulas en cada momento. Para formular este proceso de decisión se define un vector de $1 \times (N_c \times N^\circ \text{ periodos simulados})$ en el que cada conjunto de 6 elementos representa la combinación a aplicar en ese instante del experimento:

$$\delta(t) \in \{\delta_1, \dots, \delta_{N_c}\} \quad (4)$$

Donde:

$$\delta_i = [0 \dots 0 \ 1 \ 0 \dots 0]^T \quad (5)$$

En este vector cada valor de $\delta(t)$ solo puede tomar los valores 0 o 1 y solo uno de los N_c elementos que lo componen puede ser igual a 1 a la vez.

Habiendo definido este vector de decisión podemos establecer el vector de flujo entrante en los tanques en un momento dado como:

$$f_T(t) = F_T \cdot \delta(t)$$

El vector de caudal de salida de las bombas de llenado se determina de forma análoga:

$$f_P(t) = F_P \cdot \delta(t)$$

De igual forma pasa con el vector de potencia consumida por las bombas:

$$p(t) = P \cdot \delta(t)$$

Para obtener el coste total en dinero que supone este consumo se deben establecer también tarifas eléctricas. Se han establecido una para cada bomba de llenado.

$$T(t) = [T_1 \ T_2] \quad (6)$$

El coste para un periodo de tiempo determinado puede ser expresado como:

$$J = \frac{1}{60} \int T(t) \cdot P \cdot \delta(t) dt \quad (7)$$

La ecuación que expresa el balance de los tanques es:

$$\dot{V}_j = f_{T,j}(t) - f_{O,j}(t) \quad (8)$$

Donde $f_{T,j}(t)$ y $f_{O,j}(t)$ son los flujos de entrada y salida de agua de los tanques respectivamente en litros por minuto. Se puede congrega todas las ecuaciones de los tanques en forma matricial. Como se puede observar a continuación:

$$\dot{V} = \begin{bmatrix} \dot{V}_j \\ \dots \\ \dot{V}_{N_t} \end{bmatrix} = \begin{bmatrix} f_{T,1}(t) \\ \dots \\ f_{T,2}(t) \end{bmatrix} - \begin{bmatrix} f_{O,1}(t) \\ \dots \\ f_{O,2}(t) \end{bmatrix} = f_T(t) - f_O(t) \quad (9)$$

Sin embargo, el flujo de salida futuro de los tanques, $f_O(t)$, no se conoce a priori y por lo tanto debe utilizarse una predicción del flujo de salida, $\hat{f}_O(t)$, para calcular la evolución del volumen. La ecuación del volumen de los tanques es la siguiente:

$$\dot{V} = \frac{1}{60} (f_T(t) - \hat{f}_O(t)) \quad (10)$$

A la hora de aplicar estas fórmulas en el código estas deben ser discretizadas con un periodo de tiempo “ h ”, el cual para este proyecto se ha elegido 1.3 segundos simulados. Es simulado puesto que el experimento tiene un tiempo interno, determinado por un temporizador en el que cada ciclo de este supone 1 periodo simulado. El periodo de temporización actual (1.3), es el resultado de un ajuste experimental en la búsqueda de

realizar la escala temporal de las pruebas experimentales lo más pequeña posible. Si bien en los cálculos iniciales se planeó hacer la escala a 1:60, donde 1 segundo de la planta de laboratorio equivale a 60 segundos de una instalación real, esto no pudo hacerse finalmente ya que un periodo menor a 1.3 segundos produce que el programa no pueda acabar de ejecutarse correctamente y la ejecución del código falla.

Para poner un ejemplo de esta discretización, los vectores $T(t)$ y $\delta(t)$ pasarían a ser vectores discretizados tales como $T[k] = T(t = kh)$ y $\delta[k] = \delta(t = kh)$.

El índice de coste expresado anteriormente se transforma en:

$$J = \frac{h}{60} \sum T[k] \cdot P \cdot \delta[k] \quad (11)$$

La discretización de la ecuación de los tanques resulta en la siguiente expresión:

$$V[k+1] = V[k] + \frac{h}{60} (f_T[k] - \hat{f}_O[k]) \quad (12)$$

Siendo $V[k] = V(t = kh)$ y $\hat{f}_O[k] = \frac{1}{h} \int_{kh}^{(k+1)h} \hat{f}_O(t) dt$.

Las restricciones del volumen máximo y mínimo se pueden expresar como:

$$V_{i,min} \leq V_i[k] \leq V_{i,max}, i = 1, \dots, N_t \quad (13)$$

O como:

$$V_{min} \leq V[k] \leq V_{max} \quad (14)$$

Donde V_{min} y V_{max} son matrices conteniendo los valores máximos y mínimos permitidos en los dos tanques del proyecto.

6.6.2 REDUCCIÓN DE LA CARGA COMPUTACIONAL DE LA OPTIMIZACIÓN

A esta restricción de mantener el volumen de los tanques entre su máximo y mínimo permitido debe añadirse otra que consiste en que los tanques deben acabar el día con el mismo volumen de agua con el que empezaron. Si no se implementase esta restricción la optimización tendería a vaciar los tanques completamente.

Teniendo en cuenta las ecuaciones y vectores discretizados anteriormente, además de tomar un horizonte de minimización $t_m = k_m h$, el problema de minimización puede plantearse como sigue:

$$\begin{array}{ll} \min_{\delta} & \frac{h}{60} \sum_{k=1}^{k_m} T[k] \cdot P \cdot \delta[k] \\ \text{Sujeto a} & \end{array} \quad (15)$$

$$\begin{aligned}
 V_{min} &\leq V(0) + \frac{h}{60} \sum_{j=1}^k (f_T[j] - \hat{f}_O[j]) \geq V_{max} \\
 \sum_{j=1}^{k_m} (f_T[j] - \hat{f}_O[j]) &\geq 0 \\
 \sum_{i=1}^{N_c} \delta_i[k] &= 1, \delta_i[k] \in \{0,1\} \\
 k &= 1, \dots, k_m
 \end{aligned}$$

Las variables de decisión del problema de optimización son los elementos del vector $\delta[k]$, los cuales solo pueden tomar los valores 0 o 1. El número de variables de decisión que se tienen es $N_c k_m$ y el número de restricciones $(2N_c + 2N_t + 1)k_m + N_t$. Para un horizonte de predicción de 1 día y, teniendo en cuenta que el periodo de discretización es de 1.3 segundos y se corresponde a 1 minuto de tiempo escalado del experimento, el número de variables sería $1440N_c$ y el número de restricciones $1440(2N_c + 2N_t + 1)k_m + N_t$ lo cual resulta en un número inmenso de restricciones que pueden suponer una gran carga computacional. Además, los problemas de optimización mixta-entera de por sí ya suponen una gran carga computacional y a eso hay que sumarle que la optimización se realiza muchas veces durante el día. Esto es así ya que el caudal de demanda $\hat{f}_O[k]$ es una estimación y puede llegar a variar significativamente del predicho, además de porque existe la posibilidad de mejorar los resultados de la optimización teniendo en cuenta los valores de caudal de demanda que se van registrando a lo largo del día. En el proyecto se ha hecho uso de un intervalo de 10 minutos entre optimizaciones, lo cual supone que tan solo las variables de decisión correspondientes a los 10 primeros minutos de la optimización se aplicarían realmente y el resto se descartaría.

Por ello para rebajar el coste computacional se ha planteado establecer dos periodos de discretización, uno más preciso para los primeros instantes “h” y otro más grueso para el resto, $h_M = Lh$. Otra medida de la que se hace uso para reducir la carga computacional es el permitir valores reales del vector δ en el periodo grueso, pero con la condición que la suma de sus elementos de 1. De esta forma los valores del vector representan el porcentaje de tiempo que el sistema debe estar en cada una de las combinaciones posibles e implica una reducción de la severidad de las restricciones al no tener que establecer una única combinación para todo un periodo largo. Por otra parte, el coste de optimización también desciende al no haber tantas variables enteras.

Para expresar el problema de optimización de forma más compacta se ha pasado a forma matricial, además de definir el horizonte temporal corto $t_m = k_m h$ y el horizonte temporal largo $t_M = t_m + (k_M - k_m)Lh$. Además, también se definen los vectores:

$$T[k] = \begin{cases} T(t = kh) & \text{si } k \leq k_m \\ T(t = k_m h + (k - k_m)Lh) & \text{si } k > k_m \end{cases} \quad (16)$$

$$\hat{f}_o[k] = \begin{cases} \frac{1}{h} \int_{kh}^{(k+1)h} \hat{f}_o(t) dt & \text{si } k \leq k_m \\ \frac{1}{Lh} \int_{k_m h + (k+1-k_m)Lh}^{k_m h + (k-k_m)Lh} \hat{f}_o(t) dt & \text{si } k > k_m \end{cases} \quad (17)$$

Y los siguientes vectores y matrices:

$$\Delta = \begin{bmatrix} \delta[1] \\ \vdots \\ \delta[k_m] \end{bmatrix}_{k_M N_c \times 1} \quad (18)$$

$$T = \frac{h}{60} [T[1] \cdots T[k_m] \quad T[k_m + 1]L \cdots T[k_m]L] \quad (19)$$

$$P = \begin{bmatrix} P & 0 & \cdots & 0 \\ 0 & P & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & P \end{bmatrix}_{k_M N_p \times k_M N_c} \quad (20)$$

$$F_k = \begin{bmatrix} \underbrace{F_T \cdots F_T}_{k_m} & \underbrace{LF_T \cdots LF_T}_{k-k_m} & 0 \cdots 0 \end{bmatrix}_{N_t \times k_M N_c} \quad (21)$$

$$I_k = \begin{bmatrix} \underbrace{0 \cdots 0}_{(k-1)N_c} & \underbrace{1 \cdots 1}_{N_c} & 0 \cdots 0 \end{bmatrix}_{1 \times k_M N_c} \quad (22)$$

$$\hat{F}_o[k] = \sum_{j=1}^{k_m} \hat{f}_o[j] + L \sum_{j=k_m+1}^k \hat{f}_o[j] \quad (23)$$

Habiendo realizado estas definiciones se pasa a formular el problema de minimización como:

$$\min_{\Delta} \quad TP\Delta \quad (24)$$

Sujeto a

$$F_k \Delta \geq 60 \frac{V_{min} - V(0)}{h} + \hat{F}_o[k], \quad k = 1, \dots, k_M$$

$$F_k \Delta \leq 60 \frac{V_{max} - V(0)}{h} + \hat{F}_o[k], \quad k = 1, \dots, k_M$$

$$F_{k_M} \Delta \geq \hat{F}_o[k_M]$$

$$I_k \Delta = 1, \quad k = 1, \dots, k_M$$

$$\Delta[j] \in \{0,1\} \in \mathbb{N}, \quad j = 1, \dots, k_m N_c$$

$$\Delta[j] \in \{0,1\} \in \mathbb{R}, \quad j = k_m N_c + 1, \dots, k_M N_c$$

El número de variables de decisión enteras en este nuevo planteamiento es $N_c k_m$ y el número de variables de decisión en números reales es $N_c (k_M - k_m)$. Mientras que el número de restricciones pasa a ser $(2N_c + 2N_t + 1)k_M + N_t$.

6.6.3 RESTRICCIONES DE VOLUMEN ADICIONALES Y LIMITACIÓN DE LAS CONMUTACIONES

Un problema que presenta la formulación de la optimización hasta ahora es que el volumen real de los tanques podría llegar a desbordarse de los límites establecidos debido a la falta de precisión y fiabilidad de la predicción del flujo de caudal de salida de los tanques. Si eso ocurre la optimización no se puede resolver, debido a que empieza en un punto que incumple las restricciones y no puede encontrar una alternativa. Para solucionar esto se han aplicado dos medidas. La primera es la de fijar los límites máximos y mínimos con un margen de seguridad respecto a los límites reales de los tanques y la segunda es la de permitir que los volúmenes máximos y mínimos sean rebasados en cierta cantidad durante los primeros instantes después de la optimización.

Para aplicar esta solución se estableció un vector que varía en el tiempo, teniendo unos límites más amplios en los instantes iniciales. Para las restricciones de volumen máximo este es:

$$V_{sup}[k] = \begin{bmatrix} V_{sup,1}[k] \\ V_{sup,2}[k] \end{bmatrix} \quad (25)$$

Donde

$$V_{sup,i}[k] = \begin{cases} V_{max}(i) & \text{si } V_i(0) \leq V_{max}(i), \forall k \\ V_{max}(i) & \text{si } V_i(0) > V_{max}(i), \forall k > k_m \\ \frac{V_i(0)(k_m - k) + V_{max}(i)k}{k_m} & \text{si } V_i(0) > V_{max}(i), \forall k \leq k_m \end{cases} \quad (26)$$

Mientras que para las restricciones de volumen mínimo se tiene:

$$V_{inf}[k] = \begin{bmatrix} V_{inf,1}[k] \\ V_{inf,2}[k] \end{bmatrix} \quad (27)$$

Siendo

$$V_{inf,i}[k] = \begin{cases} V_{min}(i) & \text{si } V_i(0) \geq V_{min}(i), \forall k \\ V_{min}(i) & \text{si } V_i(0) < V_{min}(i), \forall k > k_m \\ \frac{V_i(0)(k_m - k) + V_{min}(i)k}{k_m} & \text{si } V_i(0) < V_{min}(i), \forall k \leq k_m \end{cases} \quad (28)$$

En las ecuaciones [24] V_{min} y V_{max} deberían ser sustituidos por V_{inf} y V_{sup} respectivamente.

Otro contratiempo que se tiene es que la solución óptima que se obtiene conllevaría que las bombas y las válvulas conmutasen constantemente. Lo cual supondría un gran gasto económico y un desgaste significativo de la vida útil de estos componentes hidráulicos.

Si se consideran las conmutaciones como cambios en el vector $\delta[k]$ durante el periodo de optimización, se puede desarrollar una ecuación que interrelacione el número de cambios con la matriz Δ . Sin embargo, ya que los elementos de δ pueden tomar valores reales en los periodos de optimización gruesos, el número de optimizaciones solo puede ser contado en los periodos cortos donde los valores de δ solo pueden ser enteros.

Si definimos las siguientes matrices:

$$I_N^+ = \begin{bmatrix} \overbrace{0 \cdots 0}^{N_c} & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \cdots & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}_{(k_m-1)N \times k_m N}$$

$$I_N^- = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & \underbrace{0 \cdots 0}_{N_c} \end{bmatrix}_{(k_m-1)N \times k_m N}$$

$$Y_{N_c} = I_N^+ - I_N^-$$

El número de cambios en $\delta[k]$ durante los periodos cortos puede expresarse como:

$$\text{sum} \left(\text{abs} \left(Y_{N_c} [\delta_{ant}; \Delta(1:(k_m - 1)N_c)] \right) \right) \quad (29)$$

Esta expresión tiene en cuenta como posible conmutación si existe un cambio entre el ultimo valor aplicado de Δ en la optimización pasada y el primer valor de Δ en la optimización actual.

Existen dos métodos para reducir el número de conmutaciones, añadir el número de cambios de δ al índice de coste o bien establecer este límite como una restricción. Para el primer caso el problema de optimización tendría las mismas restricciones que en [24] y únicamente el índice de coste sería distinto:

$$J = J + \alpha_c \text{sum} \left(\text{abs} \left(Y_{N_c} [\delta_{ant}; \Delta(1:(k_m - 1)N_c)] \right) \right) \quad (30)$$

El factor de ponderación α_c debe seleccionarse con cautela manteniendo un equilibrio entre minimizar el coste o el número de conmutaciones.

La segunda forma de disminuir las conmutaciones mediante el uso de restricciones, es posible de aplicar, pero no práctico ya que carga en exceso el ordenador. Además, se puede dar el caso de que, aunque se restrinjan las conmutaciones a 2, y sabiendo que la optimización se ejecuta cada 10 minutos es posible que las 2 conmutaciones permitidas se den en esos 10 minutos iniciales. Eso conlleva que al final del día se tuviesen $2 \times 6 \times 24 = 288$ conmutaciones en un día.

Como se requiere como restricción que el volumen final sea igual al inicial, la optimización depende del momento del día en el que se ejecuta y del nivel de los tanques en dicho instante. Para prevenir esto y también como medida de seguridad para contrarrestar imprevistos que surjan durante el día, se establece la restricción secundaria de que al final del periodo tarifario barato (8 am) el volumen de los tanques se encuentre en su máximo permitido.

Para conseguir tener el nivel máximo en los tanques en un momento determinado y debido a la discretización del problema se ha propuesto un valor " k_v " como el índice más cercano al momento en el tiempo en el que se desea fijar el volumen. Si se establece

“ t_i ” como el momento en el tiempo en el que se ejecuta la optimización y “ t_v ” al tiempo en el que se desea tener el máximo volumen, el valor de k_v es:

$$k_v = \begin{cases} \text{round}\left(\frac{t_v - t_i}{h}\right) & \text{si } t_v - t_i \leq k_m h \\ k_m + \text{round}\left(\frac{t_v - t_i - k_m h}{Lh}\right) & \text{si } t_v - t_i > k_m h \end{cases} \quad (31)$$

Sumándole este término al índice de coste este quedaría como:

$$J = J + \alpha_v \left(\hat{F}_O[k_v] - F_{k_v} \Delta + 60 \frac{V_{obj} - V(0)}{h} \right) \quad (32)$$

Donde α_v es un vector de los factores de ponderación para cada tanque y V_{obj} es un vector con los volúmenes a alcanzar en el instante t_v . Con este método el conseguir o no el volumen máximo en el momento deseado depende mucho de los factores de corrección elegidos, la alternativa es añadir la siguiente restricción:

$$F_{k_v} \Delta \geq \hat{F}_O[k_v] + 60 \frac{V_{obj} - V(0)}{h} \quad (33)$$

Sin embargo, aplicar esta restricción puede resultar en que la optimización sea irresoluble, especialmente cuando el momento en el que se ejecuta la optimización es cercano al momento en el que se fija el volumen máximo. Es decir, si k_v es pequeño.

6.6.4 ESTADO FINAL DEL PROBLEMA DE OPTIMIZACIÓN

En última instancia el problema de optimización cambiará dependiendo de qué condiciones se ponen como restricciones o en el índice de coste. A continuación, se presenta el planteamiento final del problema de optimización.

Se define:

$$J_0 = TP\Delta \quad (34)$$

$$J_c = \alpha_c \text{sum} \left(\text{abs}(Y_{N_c}[\delta_{ant}; \Delta(1:(k_m - 1)N_c)]) \right) \quad (35)$$

$$J_v = \alpha_v \left(\hat{F}_O[k_v] - F_{k_v} \Delta + 60 \frac{V_{obj} - V(0)}{h} \right) \quad (36)$$

Las siguientes serían las restricciones comunes a los distintos planteamientos:

$$F_k \Delta \geq 60 \frac{V_{inf}[k] - V(0)}{h} + \hat{F}_O[k], \quad k = 1, \dots, k_M \quad (37)$$

$$F_k \Delta \leq 60 \frac{V_{sup}[k] - V(0)}{h} + \hat{F}_O[k], \quad k = 1, \dots, k_M$$

$$F_{k_M} \Delta \geq \hat{F}_O[k_M]$$

$$I_k \Delta = 1, \quad k = 1, \dots, k_M$$

$$\Delta[j] \in \{0,1\} \in \mathbb{N}, \quad j = 1, \dots, k_m N_c$$

$$\Delta[j] \in \{0,1\} \in \mathbb{R}, \quad j = k_m N_c + 1, \dots, k_M N_c$$

La restricción adicional de fijar un volumen objetivo es:

$$C_V = F_{k_V} \Delta \geq \hat{F}_O[k_V] + 60 \frac{V_{obj} - V(0)}{h} \quad (38)$$

Por lo tanto, las posibles combinaciones de objetivos y restricciones son:

- 1) Minimizar J_0 y J_c sujeto a las restricciones comunes y a C_V .
- 2) Minimizar J_0, J_c y J_v sujeto a las restricciones comunes únicamente.

Tanto el limitar el número de conmutaciones como el fijar un volumen determinado en un cierto instante se puede implementar repercutiendo en el índice de coste o como una restricción del problema. Siendo el primero una solución más blanda o permisiva que la segunda y por lo tanto suponer menor carga computacional y mayor probabilidad de encontrar solución a la optimización. Teniendo esto en cuenta, en la implantación final se ha optado por dar preferencia a la primera opción planteada de combinación de restricciones, solo utilizando la segunda en caso de que el problema sea irresoluble, teniendo por lo tanto una función de salvaguarda en caso de error.

6.6.5 PREDICCIÓN DEL FLUJO DE SALIDA

Para que la optimización pueda generar la gestión óptima de las bombas y válvulas correspondientes, necesita de una predicción del caudal de salida \hat{F}_O . A continuación, se explicará cómo se estima dicha predicción.

El algoritmo de predicción utilizado hace que la predicción de \hat{F}_O en el momento 24 horas después del actual sea función de la medida actual y de la medida tomada justo 6 días atrás, teniendo en cuenta factores como fiestas o la diferencia entre un día laborable y el fin de semana. Si se tiene un determinado periodo h , en segundos, y el flujo de salida medido en el instante $k = \frac{t}{h}$ es $f_O[k]$, entonces la predicción para ese instante se actualizaría de la siguiente manera:

$$\hat{f}_O[k] = f_O[k] \quad (39)$$

De esta forma el sistema de predicción se actualiza de forma que las predicciones de 24 horas y 1 semana en el futuro se mejoran.

Para complementar esta dependencia de las predicciones futuras de los valores registrado anteriormente, se definen dos factores de ponderación β_d y β_w , tal que la sumas de ambos de 1. Aplicando esta ponderación, la predicción 24 horas en el futuro quedaría así:

$$\hat{f}_o \left[k + 1440 \cdot \frac{60}{h} \right] = \beta_d \hat{f}_o[k] + \beta_w \hat{f}_o \left[k - 6 \cdot 1440 \cdot \frac{60}{h} \right] \quad (40)$$

Los valores de β_d y β_w dependen del día de la semana, siendo de lunes a jueves el valor de $\beta_d = 1$ ya que entre estos días las demandas de caudal se suponen muy parecidas, habiendo cambios importantes en viernes y domingo al marcar el paso de laborable a fin de semana y viceversa. En la Tabla 2 se recogen algunos ejemplos de factores de ponderación que se podrían aplicar:

Día de la semana	β_d	β_w
Lunes	1	0
Martes	1	0
Miércoles	1	0
Jueves	1	0
Viernes	0.1	0.9
Sábado	0.8	0.2
Domingo	0.1	0.9

Tabla 2. Ejemplo de factores de ponderación para distintos días

Sin embargo, para la puesta en práctica se han realizado experimentos de máximo 2 días virtuales de duración debido a la imposibilidad de escalarlos temporalmente en mayor medida. Por lo tanto, únicamente la predicción 24 horas en el futuro se llega a realizar.

Por otra parte, la predicción del flujo de salida puede ser mejorada para el futuro cercano corrigiéndola con los datos actuales que se han registrado. Esto es así ya que, si la predicción para el siguiente periodo es $\hat{f}_o[k+1] = 100$ y el valor registrado en este instante es $f_o[k] = 200$, el valor real de $f_o[k+1]$ se acabará pareciendo más a 200 que a 100. Por ello se define una constante de tiempo t_f para actualizar las predicciones cercanas en el tiempo, ponderándolas exponencialmente con el error de predicción actual.

$$\hat{f}_o[k+j] = \hat{f}_o[k+j] + e^{\frac{-jh}{t_f}} (f_o[k] - \hat{f}_o[k^-]) \quad (41)$$

Siendo $\hat{f}_o[k^-]$ la predicción en el instante k antes de que la medida $f_o[k]$ haya sido tomada.

6.6.6 FLUJO DE SALIDA REAL

Por una parte, se tiene la predicción del caudal que utiliza la optimización para realizar sus cálculos y resolver el problema de optimización y por otra está el caudal de demanda real que se utiliza como referencia en las bombas de vaciado de los tanques. Este caudal real de salida no es conocido por la optimización. Su implementación se realiza en el PLC, mientras que la optimización se ejecuta en el PC. Como se avanzaba anteriormente esta curva de caudal sigue una serie de Fourier ajustada a un sistema de suministro de agua real, la cual se presenta a continuación:

$$Q_{demand} = (a_0 + a_1 \cos(wt) + a_2 \cos(2wt) + a_3 \cos(3wt) + b_1 \sin(wt) + b_2 \sin(2wt) + b_3 \sin(3wt)) \cdot \varepsilon \cdot (1 + random) \quad (42)$$

Siendo

$$f_o = Q_{demand}$$

$$w = \frac{2\pi}{1440}, a_0 = 97.64, a_1 = -36.18, a_2 = -4.421, a_3 = 14.34, b_1 = 1.699, b_2 = -17.57, b_3 = 4.85$$

El factor ε es una variable obtenida experimentalmente para desplazar proporcionalmente la curva de caudal a los rangos deseados, mientras que el último término se utiliza para añadirle un cierto grado de aleatoriedad aparte de la imprecisión y fluctuación propia derivada de los componentes. El rango de caudal al que se ha ajustado las curvas de demanda para los experimentos se sitúa con un máximo de 2 litros por minuto y un mínimo de demanda de alrededor de los 0.6 litros por minuto.

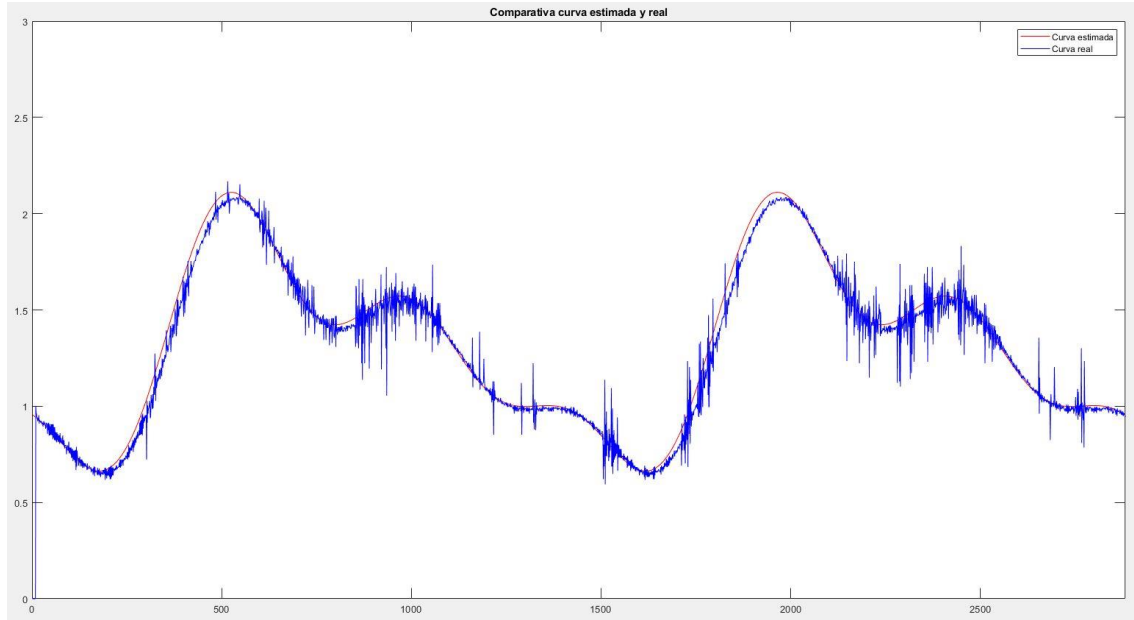


Figura 16. Comparativa curva de caudal estimado sin actualizar y caudal demandado real

La Figura 16 muestra una comparación del caudal predicho respecto al caudal medido a lo largo de dos días virtuales de experimento. La curva de caudal predicho se ha realizado a partir del algoritmo descrito anteriormente. Esta curva, conforme pasa el tiempo, se actualiza hasta adquirir la misma forma que el caudal de demanda real, sin embargo, en esta gráfica se ha representado el caudal predicho sin actualizar para observarlos por separado. La curva de estimación de la demanda tiene un componente que le aporta aleatoriedad y con el cual se consigue que se parezca a la curva real, pero con variaciones en proporcionalidad y rizado. Es así para que sea con las actualizaciones que se dan durante el experimento que la predicción se consiga mejorar.

6.7 CÓDIGO DE MATLAB

En esta sección se presentará y explicará brevemente el código utilizado en el programa Matlab para el desarrollo de este proyecto.

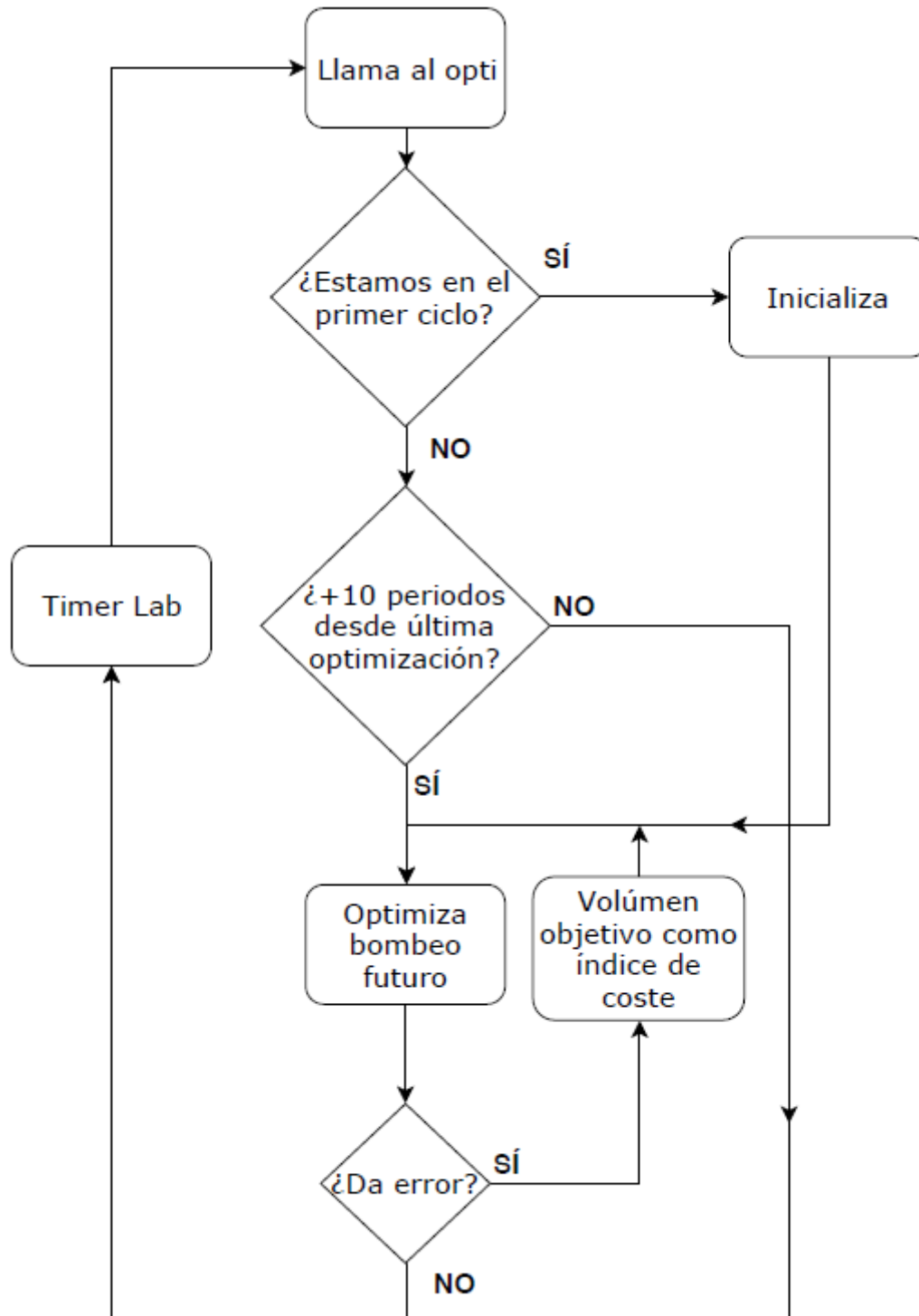


Figura 17. Estructura de funcionamiento del código en Matlab

El código implementado en Matlab está dividido en tres programas y una función:

- Llama al opti. Este es el programa principal desde el que se llama a todos los demás programas y a la función.
- Inicializa. Es el programa inicial que se ejecuta únicamente una vez durante cada experimento y establece todos los valores esenciales.
- Optimiza bombeo futuro. Es el programa en el que se ejecuta la optimización y devuelve su resultado al programa principal.
- Timer Lab. Es la función de temporización que permite escalar temporalmente los experimentos y determina por lo tanto el periodo de temporización.

En la Figura 17 podemos observar el flujo de ejecución entre los distintos programas y la función, además de alguna de las decisiones más importantes que los relacionan.

6.7.1 PROGRAMA: INICIALIZA

Esta primera sección del programa inicializa que se observa en la figura 18 se establecen los valores de las variables principales del problema de optimización tales como el valor de las potencias de las bombas, los caudales de las bombas en cada configuración hidráulica o los volúmenes mínimos y máximos permitidos en los tanques.

```
yal mip clear

P11 = 15;
P21 = 20;
P22 = 10;

P = [0 P11 0 0 P11 P11; 0 0 P21 P22 P22 P21];

Sp = [0 1 0 0 1 1; 0 0 1 1 1 1];

X=3;

W = [X X 1 0 0 1; X X 0 1 1 0];

Q11 = 2;
Q22 = 1.9;
Q21 = 1.7;

q_1 = [0 Q11 Q21 0 Q11 Q11+Q21];
q_2 = [0 0 0 Q22 Q22 0];

FT = [ q_1; q_2];

Fp = [0 Q11 0 0 Q11 Q11; 0 0 Q21 Q22 Q22 Q21];

Nc=6;
Nv=2;
Np=2;
Nt=2;

Vmin=[1;1];
Vmax=[4.5;4.5];
Vinicial=Vmin+0.8*[1;1].*(Vmax-Vmin);

combinaciones=6;
```

Figura 18. Inicializa - Bloque de código 1

El siguiente bloque establece los periodos tarifarios para cada bomba.

```
T1f=[0.063*ones(1,480),0.079*ones(1,540-480),0.1*ones(1,900-540),0.079*ones(1,1440-900)];
T2f=[0.066*ones(1,480),0.088*ones(1,600-480),0.099*ones(1,960-600),0.088*ones(1,1440-960)];
```

Figura 19. Inicializa - Bloque de código 2

Por otra parte, también se establece la curva de caudal de salida estimado que se usará en la optimización para estimar el comportamiento de los tanques y encontrar la solución óptima al problema de gestión de las bombas y válvulas.

```
a0 = 97.64; a1 = -36.18; b1 = 1.699; a2 = -4.421; b2 = -17.57; a3 = 14.34; b3 = 4.85;
w = 0.2618;
a=[a1;a2;a3];b=[b1;b2;b3];
nf=3;
FRACAUDAL=[0.0133*0.9;0.0133];
for iti=1:ndias*1441
    Fbase = a0 + cos((iti+tini)/60*w*(1:nf))*a + sin((iti+tini)/60*w*(1:nf))*b;

    Fsalidaestimada(:,iti)=[Fbase;Fbase].*FRACAUDAL*(1+rand(1)*0.1);

end
```

Figura 20. Inicializa - Bloque de código 3

6.7.2 PROGRAMA: LLAMA AL OPTI

Para comenzar se definen las variables que se van a compartir entre los diferentes programas y el temporizador como variables globales de forma que se pueda acceder a ellas y modificarlas desde cualquier parte del proyecto. Esto se hace ya que las variables sí que se comparten entre programas, pero no con la función de temporización al usar otro hilo de ejecución.

```
clear all;
global bandera;

bandera = 1;

global direcc_nivel1;
global direcc_nivel2;
global Volmedido;
global Qmedido;
global Qbombeo;
global direcc_bol;
global direcc_bo3;
global Valvula;
global Deltaaplicada;
global Fp;
global Nc;
global Fsalidaestimada;
global perexpoQ;
global direcc_caudal2;
global direcc_caudal4;
global direcc_ev4;
global direcc_ev3;
global W;
global Ferror;
global t;
```

Figura 21. Llama al opti - Bloque de código 1

Tras esto se definen las direcciones de memoria MODBUS del PLC WAGO que vamos a utilizar para transmitir las referencias a las bombas y válvulas y obtener las lecturas de los sensores de nivel y caudalímetros.

```
%Direcciones de memoria
direcc_perodo = 12307;
direcc_caudal_salida = 12308;
direcc_caso = 12309;
direcc_tiempoMatlab = 12310;

direcc_bo1 = 12289;
direcc_bo2 = 12290;
direcc_bo3 = 12291;
direcc_bo4 = 12292;

direcc_caudal1 = 12293;
direcc_caudal2 = 12294;
direcc_caudal3 = 12295;
direcc_caudal4 = 12296;

direcc_nivel1 = 12297;
direcc_nivel2 = 12298;

direcc_ev1 = 12299;
direcc_ev2 = 12300;
direcc_ev3 = 12301;
direcc_ev4 = 12302;
```

Figura 22. Llama al opti - Bloque de código 2

En la Figura 23 se llama al programa inicializa explicado en el subapartado anterior. Se define el número de días de virtuales (ndías), en nuestro caso 2, y se establecen los tiempos inicial y total del experimento en número de periodos.

También se establece el volumen objetivo y el periodo en el que se quiere, a las 8 de la mañana, el final del periodo tarifario barato.

En el último párrafo de este bloque se establecen diversas variables vistas ya en el apartado de optimización como el valor del periodo ("h" en el apartado de optimización), km, L, o cada cuantos periodos se debe ejecutar la optimización (periodooptim = 10).

```
inicializa;

ndias=2;
tinicial=10;
tini=tinicial;
tsimttotal=ndias*1440+tini;
t=tini;

Vmaxobj=Vmin+0.95*(Vmax-Vmin);
tVmax=480;

periodooptim=10;
periodo=1.3;
km=30;
L=30/periodo;
KM=(1440-km*periodo)/L/periodo+km;
```

Figura 23. Llama al opti - Bloque de código 3

Tras esto se realiza una primera optimización y se establecen las condiciones para esta, como por ejemplo las condiciones que se utilizan en el índice de coste, que restricciones del problema de optimización implementar o los factores de ponderación a aplicar en las distintas restricciones. Tanto esta primera optimización como el resto más adelante están envueltas en una estructura condicional en la que si ocurre un fallo en la optimización este queda registrado en el vector “infeas”. En el caso de error se establece el volumen objetivo a las 8 de la mañana como un término más del índice de coste para facilitar la convergencia de una solución óptima. Tras la resolución de esta segunda optimización se restablecen las condiciones iniciales del problema.

```

excV=0.1;
alfa=0.1;
alfaV=0.001;
soloJ=false;
JVmaxobj=false;
Jnumconm=true;
resfinal=true;
ResVmaxobj=true;
Resnumconm=false;
if soloJ==true
    JVmaxobj=true;
    ResVmaxobj=false;
end
optimiza_bombeo_futuro
if sol.problem==1
    infeas=[infeas,1];
    JVmaxobj=true;
    ResVmaxobj=false;
    optimiza_bombeo_futuro

    if sol.problem==1
        infeas2 = [infeas2,1];
    else
        infeas2=[infeas2,0];
    end

else
    infeas=[infeas,0];
end
JVmaxobj=false;
ResVmaxobj=true;
if soloJ==true
    JVmaxobj=true;
    ResVmaxobj=false;
end

```

Figura 24. Llama al opti - Bloque de código 4

Una vez acabada la primera optimización, los resultados de los 10 primeros periodos que son los que se van a aplicar finalmente, se guardan en una variable de nombre Deltaaplicada, para su posterior uso.

```

for ik=1:periodooptim
    Deltaaplicada(t*Nc+Nc*(ik-1)+1:t*Nc+Nc*ik)=value(Deltab((ceil(ik/periodo)-1)*Nc+1:
Nc*ceil(ik/periodo)));
end

```

Figura 25. Llama al opti - Bloque de código 5

El valor de la última delta aplicada antes de la siguiente optimización también se guarda (Deltaini), ya que si no se tuviera en cuenta al inicio de cada optimización se

contaría una conmutación, lo cual a lo largo de un día entero serían demasiadas. También se acumulan las conmutaciones y el coste de esos 10 primeros periodos de operación.

```
Deltaini=Deltaaplicada(t*Nc+Nc*periodooptim-Nc+1:t*Nc+Nc*periodooptim);
conmtot=conmtot+conmbo;
Jtot=Jtot+Jo;
```

Figura 26. Llama al opti - Bloque de código 6

A estas alturas se configura la función de temporización “Timer_Lab” de la siguiente forma:

```
periodo_codesys = 1.3;
tv = timer('StartDelay', 0, 'Period', periodo_codesys, 'ExecutionMode', 'fixedRate');
tv.TimerFcn = @(~,thisEvent)Timer_Lab;
```

Figura 27. Llama al opti - Bloque de código 7

En la Figura 28 se inicia la comunicación del PC con el PLC mediante el objeto “m”, el cual establece la conexión MODBUS. Se envían los valores de referencia de las bombas y los estados de las válvulas obtenidos en la optimización anterior. Además, también se leen los niveles de agua de los tanques (Volmedido) y el caudal de salida de estos (Qmedido).

```
global comienzo;
m = modbus('tcpip', '192.168.1.3', 502);

Nivell_PLC = read(m, 'holdingregs', direcc_nivell,1);
Nivell_Litros = Nivell_PLC*(4.867/32761);
Nivel2_PLC = read(m, 'holdingregs', direcc_nivel2,1);
Nivel2_Litros = Nivel2_PLC * (4.44/32761);
Volmedido(:,t)= [Nivell_Litros; Nivel2_Litros];
QSalida1_PLC = read(m, 'holdingregs', direcc_caudal2,1);
QSalida2_PLC = read(m, 'holdingregs', direcc_caudal4,1);
QSalida1 = QSalida1_PLC*(8/16383);
QSalida2 = QSalida2_PLC*(8/16383);
Qmedido(:,t)= [QSalida1;QSalida2];
Qbombeo(:,t)= Fp*Deltaaplicada((t-1)*Nc+1:t*Nc);
Qbombeo1 = Qbombeo(1,t);
Qbombeo2 = Qbombeo(2,t);
Qbombeo1_PLC = round(Qbombeo1*(16383/8));
Qbombeo2_PLC = round(Qbombeo2*(16383/8));
write(m,'holdingregs', direcc_bo1, Qbombeo1_PLC);
write(m,'holdingregs', direcc_bo3, Qbombeo2_PLC);
Valvula(:,t)=W*Deltaaplicada((t-1)*Nc+1:t*Nc);
```

Figura 28. Llama al opti - Bloque de código 8

Las variables del vector “Valvula” obtenidas de la optimización deben de ser tratadas ya que, como se puede observar en la Tabla 1, uno de los valores que pueden adoptar los elementos de esta variable es “X” al cual en el código se le ha dado un valor de 3. Este valor viene a decir que no importa el estado en el que se encuentren las válvulas puesto que la bomba 2 no se encuentra bombeando. Por lo tanto, se implementa un bucle “for” el cual cambia ese valor de 3 por el mismo valor que hay presente en el elemento justo anterior de forma que no se produzca una conmutación innecesaria del estado de las válvulas. Tras aplicar esta corrección se transmite al PLC el nuevo estado que deben tener las válvulas.

```

for ij=1:2
    if Valvula(ij,t)==3
        Valvula(ij,t)=Valvula(ij,t-1);
    end
end
valvula4 = Valvula(1,t);
valvula3 = Valvula(2,t);
write(m,'holdingregs' , direcc_ev4, valvula4);
write(m,'holdingregs' , direcc_ev3, valvula3);

```

Figura 29. Llama al opti - Bloque de código 9

Lo último que se realiza antes de entrar en el bucle principal de este programa es actualizar la curva de caudal estimada con los valores del caudal de salida real (Qmedido) medidos con anterioridad, además de corregir los valores del futuro cercano a partir de la ecuación [41] del apartado de la optimización.

```

Fsalidaestimada(:,1:1440)=Fsalidaestimada(:,2:1441);
Error=Qmedido(:,t)-Fsalidaestimada(:,1);

Fsalidaestimada(:,1)=Qmedido(:,t);
Fsalidaestimada(:,1441)=Qmedido(:,t);
for i=2:perexpoQ*4
    Fsalidaestimada(:,i)=Fsalidaestimada(:,i)+Error*exp(-i/perexpoQ);
end

```

Figura 30. Llama al opti - Bloque de código 10

Bucle principal

Se inicia un bucle “While” con la condición de que el tiempo sea inferior al tiempo total del experimento. El tiempo es actualizado por la función de temporización “Timer_Lab” por lo que en el primer ciclo de ejecución el tiempo aún no ha avanzado desde que se estableció al principio del código.

En el bucle las dos únicas cosas que se hacen son las optimizaciones e iniciar el temporizador en el primer periodo de ejecución del bucle. La optimización se realiza únicamente si el tiempo actual “t” es múltiplo de “periodooptim” (el cual tiene un valor de 10). En la primera parte del bucle se realiza la optimización, cuyo código es el mismo que el que se presenta en las Figuras 24, 25 y 26.

La segunda parte del bucle principal de “llama al opti” es la de iniciar la función de temporización en caso de que esta no se haya iniciado anteriormente. Para ello se usa una variable, “comienzo”, para registrar si este se ha iniciado o no. Por último, y ya fuera del bucle “While”, se realiza una última acumulación del número de conmutaciones y el índice de coste para no dejarse fuera los valores del último ciclo.

```

if isempty(comienzo)
    start(tv)
    write(m,'holdingregs' , direcc_periodo, periodo_codesys*10);
    comienzo = 1;
end

conmtot=conmtot+conmbo;
Jtot=Jtot+Jo;

```

Figura 31. Llama al opti - Bloque de código 11

6.7.3 PROGAMA: OPTIMIZA BOMBEO FUTURO

Una de las primeras cosas que se hacen en este programa es definir un bucle “for” en el que se reorganizan los vectores de tarifas, potencias, flujos de entrada a los tanques y flujos de salida de estos en matrices. Los valores tratados se corresponden a los periodos de optimización finos.

```
for i=1:km
    T=[T,periodo/60*T1(1,(i-1)*periodo+1),periodo/60*T2(1,(i-1)*periodo+1)];
    Pg=blkdiag(Pg,P);
    I=blkdiag(I,ones(1,Nc));
    Ff=[Ff,FT];
    F=[F;[Ff,zeros(Nt,KM*Nc-length(Ff))]];

    S=blkdiag(S,Sp);
    if (i==1)

        Fo=Fsalidaestimada(:,1);
    else

        Fo=[Fo,sum(Fsalidaestimada(:,1:i*periodo))'/periodo];
    end
end
```

Figura 32. Optimiza bombeo futuro - Bloque de código 2

De igual manera ocurre en el código de la Figura 33, con la única diferencia de que se definen los valores de estas matrices correspondientes a los periodos de optimización gruesos.

```
for i=km+1:KM
    T=[T,periodo*L/60*T1(1,(km-1)*periodo+(i-km)*periodo*L+1),periodo*L/60*T2(1,(km-1)
    *periodo+(i-km)*periodo*L+1)];
    Pg=blkdiag(Pg,P);
    I=blkdiag(I,ones(1,Nc));
    Ff=[Ff,FT*L];
    F=[F;[Ff,zeros(Nt,KM*Nc-length(Ff))]];
    S=blkdiag(S,Sp);
    Fo=[Fo,sum(Fsalidaestimada(:,1:km*periodo))'/periodo+sum(Fsalidaestimada(:,
    km*periodo+1:km*periodo+(i-km)*L*periodo))'/periodo];
end
```

Figura 33. Optimiza bombeo futuro - Bloque de código 3

Los vectores “Deltab” y “Deltaf” serán los que guarden las variables de decisión binarias y reales respectivamente. Estas se unen para formar la variable conjunta “Delta”.

```
Deltab=binvar(km*combinaciones,1);
Deltaf=sdpvar((KM-km)*combinaciones,1);

Delta=[Deltab;Deltaf];
```

Figura 34. Optimiza bombeo futuro - Bloque de código 4

En el caso que el número de conmutaciones se ponga en el índice de coste se implementa la ecuación [30] de la optimización para reflejar la influencia de las conmutaciones en el índice de coste.

```

if (Jnumconm==true)
    J=J+alfa*sum(abs(Yb*[Deltaini;Deltab(1:length(Deltab)-Nc)]));
end

```

Figura 35. Optimiza bombeo futuro - Bloque de código 5

El código de la figura 36 define “indVmax”, el cual es el número de periodos de optimización que separan el momento en el que se inició la optimización actual y el tiempo en el que se quiere el volumen máximo objetivo, las 8 de la mañana.

```

if (JVmaxobj==true) || (ResVmaxobj==true)
    dtVmaxtini=tVmax-tini;
    if dtVmaxtini<=0 dtVmaxtini=dtVmaxtini+1440; end
    if dtVmaxtini<=km*periodo
        indVmax=round(dtVmaxtini/periodo);
    else
        indVmax=km+round((dtVmaxtini/periodo-km)/L);
    end
    if indVmax==0 indVmax=KM; end
end

```

Figura 36. Optimiza bombeo futuro - Bloque de código 6

Una vez definida esta variable se puede definir el término del volumen máximo objetivo en el índice de coste en el caso de que se requiera usar esta opción, tal y como expresa la ecuación [36]. Por otra parte, se define la restricción de que el vector “Delta” solo pueda comprender valores entre 0 y 1. Hay que tener en cuenta que este vector estaba formado por “Deltab”, que es un vector que solo puede tomar valores binarios, y “Deltaf” que es un vector que toma valores reales. Por consiguiente, esta restricción cumple su función y define una parte de las variables de decisión como binarias y otra como valores reales entre 1 y 0.

```

if (JVmaxobj==true)
    J=J+alfaV*[1 1]*((60*(Vmaxobj-Vini)/periodo+Fo(:,indVmax)-F((indVmax-1)*Nt+1:
(indVmax-1)*Nt+Nt,:)*Delta)./(Vmax-Vmin));
end
Res=[0<=Delta<=1];

```

Figura 37. Optimiza bombeo futuro - Bloque de código 7

A continuación, en la figura 38 se irá completando el vector de restricciones “Res” con el resto de condiciones del problema. En las siguientes líneas de código se comprueba si el volumen de los tanques se encuentra entre los límites permitidos, en cuyo caso se definen las restricciones de que los vectores de decisión deben de sumar 1 para cada periodo de optimización, y que los niveles de los tanques se deben mantener entre los límites antes mencionados. Estas restricciones se basan en el conjunto de ecuaciones [37] la primera y la ecuación [24] las dos últimas restricciones.

```

if (sum(Vini>Vmax)==0) && (sum(Vini<Vmin)==0)
    for i=1:KM
        Res=[Res,I(i,:)*Delta<=1];
        Res=[Res,F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(Vmin-Vini) ✓
/periodo+Fo(:,i)];
        Res=[Res,F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta<=(60/T_tiempoReal)*(Vmax-Vini) ✓
/periodo+Fo(:,i)];
    end
end

```

Figura 38. Optimiza bombeo futuro - Bloque de código 8

Continuando con el condicional “if” de la Figura 38, en el caso de que el volumen de alguno de los tanques al iniciar la optimización se encuentre fuera de los límites permitidos se definen nuevos límites y se vuelven a definir las mismas restricciones de la Figura 38, con la única diferencia que las restricciones de mantener el volumen entre los límites ahora está basada en el conjunto de ecuaciones [37] ya que los límites ya no son los originales sino que son límites expandidos para permitir reajustar los niveles y que vuelvan a estar entre el rango permitido de volumen.

```

else
    for i=1:KM
        if i>ku
            Vinf(i,:)=Vmin;
            Vsup(i,:)=Vmax;
        else
            for j=1:Nt
                if (Vini(j)<=Vmax(j))
                    Vsup(i,j)=Vmax(j);
                else
                    Vsup(i,j)=(Vini(j)*(ku-i)+Vmax(j)*i)/ku;
                end
                if (Vini(j)>=Vmin(j))
                    Vinf(i,j)=Vmin(j);
                else
                    Vinf(i,j)=(Vini(j)*(ku-i)+Vmin(j)*i)/ku;
                end
            end
        end
    end

    Res=[Res,I(i,:)*Delta<=1];
    Res=[Res,F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(Vinf(i,:)-Vini) ✓
    /periodo+Fo(:,i)];
    Res=[Res,F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta<=(60/T_tiempoReal)*(Vsup(i,:)-Vini) ✓
    /periodo+Fo(:,i)];

end
end

```

Figura 39. Optimiza bombeo futuro - Bloque de código 9

Para acabar de definir el vector de restricciones, en la figura 40, se establece la restricción de que el volumen de los tanques debe acabar como mínimo en el mismo nivel en el que empezaron, pero con una corrección por si se da el caso de que el nivel al inicio de la optimización se encuentra fuera de los límites permitidos. También se define la condición del volumen máximo objetivo a las 8 de la mañana como restricción, siendo este el estado natural de esta condición y solo implementándose en el índice de coste en el caso de que la primera optimización falle.


```

if resfinal==1
    Res=[Res,F((KM-1)*Nt+1:(KM-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(min(Vmax,Vini)-Vini)/periodo+Fo(:,KM)];
elseif advmax>0
    Res=[Res,F((KM-1)*Nt+1:(KM-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(min(Vmax,Vini)*(1-excV)-Vini)/periodo+Fo(:,KM)];
end

if ResVmaxobj==true

    Res=[Res,F((indVmax-1)*Nt+1:(indVmax-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(Vmaxobj-Vini)/periodo+Fo(:,indVmax)];
end

```

Figura 40. Optimiza bombeo futuro - Bloque de código 10

Al final del todo se inicia uno de los solvers implementados para buscar el valor óptimo del índice de coste, sujeto a las restricciones anteriores, y se guardan los valores de las conmutaciones y el coste calculados en este ciclo de optimización.

```

if strcmp(solver,'mosek')
    sol=optimize(Res,J,sdpsettings('solver',solver,'mosek.MSK_DPAR_MIO_TOL_REL_GAP',tolgap,'mosek.MSK_DPAR_MIO_MAX_TIME',maxtiempo));
else
    if strcmp(solver,'cbc')

        sol=optimize(Res,J,sdpsettings('solver',solver,'cbc.allowableFracGap',tolgap));
    else
        sol=optimize(Res,J,sdpsettings('solver','glpk','glpk.tmlim',60,'glpk.itlim',50,'glpk.lpsolver',2,'glpk.branch',1,'glpk.round',0,'glpk.btrack',1));
    end
end

conmbo=sum(abs(Ybo*[Deltaini;value(Deltab(1:periodooptim/periodo*Nc-Nc))])));
Jo=T*Pg(:,1:periodooptim/periodo*Nc)*value(Deltab(1:periodooptim/periodo*Nc));

```

Figura 41. Optimiza bombeo futuro - Bloque de código 11

6.7.4 FUNCIÓN: TIMER LAB

Esta función funciona de forma paralela al resto de programas y se ejecuta de forma cíclica con un periodo de 1.3 segundos. Su tarea principal es la de actualizar la variable temporal “t” que define el momento del día en el que se encuentra el experimento, esto se puede observar en la figura 42.

En primer lugar, se establecen las variables globales que se necesitarán para la ejecución del código y que provienen del hilo de ejecución de los programas. Tras esto se actualiza la variable “t” como se ha comentado y exactamente igual que después de la optimización inicial en “Llama al opti”, se produce la comunicación con el PLC se leen los sensores, se gestionan los actuadores y se actualiza la curva de demanda de caudal estimada (Figura 28, 29 y 30).

```
t = t+1;
```

Figura 42.Timer Lab - Bloque de código 1

6.8 CÓDIGO DEL PLC

La programación realizada en el PLC a partir del software Codesys se estructura en 2 programas:

- Demanda Caudal. El objetivo principal de este programa es definir la curva de caudal de demanda que se usará como referencia para las bombas de salida del tanque.
- Control Bombas. En este programa se pueden encontrar los controladores PID de las 4 bombas.

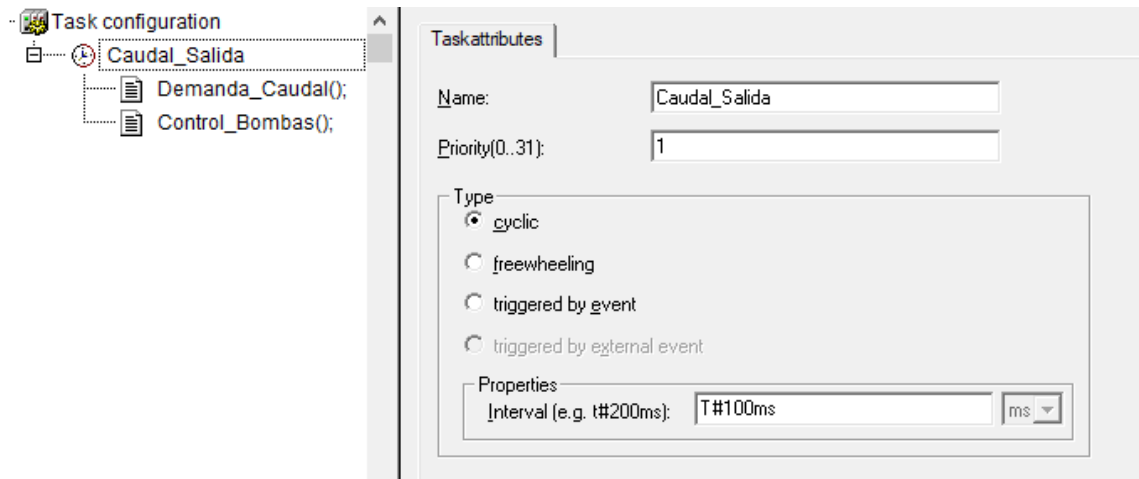


Figura 43. Configuración de tareas PLC

Estos dos programas se asignan a la tarea cíclica de periodo 100 ms que se observa en la figura 43. Ambos programas se asignan a la misma tarea puesto que no se requieren diferentes ciclos de repetición entre ellos o prioridades. El periodo que se elige es menor que la temporización del experimento y múltiplo de 10 para una fácil sincronización entre los tiempos del código en Matlab y del PLC.

6.8.1 PROGRAMA: DEMANDA CAUDAL

En primera instancia se comprueba si el experimento ha empezado, para ello se define la variable “periodo simulado” que se recibe a través de la comunicación MODBUS, si esta comunicación aún no se ha producido su valor es 0 y por lo tanto no se entra en la estructura condicional IF de la Figura 41. **Optimiza bombeo futuro - Bloque de código 11.** El valor que se recibe en “periodo simulado” esta multiplicado por 10, es decir, sería $1.3 \cdot 10 = 13$.

Dentro de la estructura IF existe otro IF que sirve para corregir posibles desviaciones entre la variable temporal “t” usada en el PLC y el tiempo del experimento usado en el PLC. Una vez fuera del IF se incrementa el valor de “t” como el resultado de la siguiente ecuación:

$$t = t + \left(\frac{1}{h \cdot 10} \right)$$

Teniendo en cuenta que “h” es 1.3 segundos y el periodo de la tarea es de 0.1 segundos, para incrementar en una unidad el valor de “t” se precisan 13 ciclos de la tarea “Caudal_Salida”.

```
START := WORD_TO_BOOL(periodo_simulado);

IF START THEN

    IF NOT (tiempo_matlab_anterior = tiempo_matlab) THEN
        t := tiempo_matlab;
    END_IF
    periodosTranscurridos := periodosTranscurridos + 1;
    t := t + (1/WORD_TO_REAL(periodo_simulado));

END_IF
```

Figura 44. Demanda Caudal - Bloque de código 1

Tras esto se configura y calcula la curva de caudal demandado que se pasará como referencia a las bombas de vaciado de los tanques, tal y como se observa en la figura 45.

```
a0 := 97.64;
a1 := -36.18;
a2 := -4.421;
a3 := 14.34;
b1 := 1.699;
b2 := -17.57;
b3 := 4.85;
W := (2*3.14)/1440;

AUX := (a0 + a1*COS(W*t) + a2*COS(2*W*t) + a3*COS(3*W*t) + b1*SIN(W*t) + b2*SIN(2*W*t) + b3*SIN(3*W*t))/30;
AUX := AUX * 820;

REF_SALIDA := REAL_TO_WORD(AUX*0.9);
REF_SALIDA4 := REAL_TO_WORD(AUX);
```

Figura 45. Demanda Caudal - Bloque de código 2

Por último, en la figura 46 se asignan las variables usadas para la comunicación MODBUS a otras variables alternativas que se pueden manipular en el programa y se guarda el tiempo o periodo en el que se encuentra la ejecución en Matlab para poder compararlo con el tiempo (t) del PLC.

```

Refbo1 := BO1_M;
Refbo2 := BO2_M;
Refbo3 := BO3_M;
Refbo4 := BO4_M;

BO1_LECTURA := BO1;
BO2_LECTURA := BO2;
BO3_LECTURA := BO3;
BO4_LECTURA := BO4;

CAU1_M := CAU_1;
CAU2_M := CAU_2;
CAU3_M := CAU_3;
CAU4_M := CAU_4;

Nivel1_M := NIVEL_1;
Nivel2_M := NIVEL_2;

EV1 := EV1_M;
EV2 := EV2_M;
EV3 := EV3_M;
EV4 := EV4_M;

tiempo_matlab_anterior := tiempo_matlab;

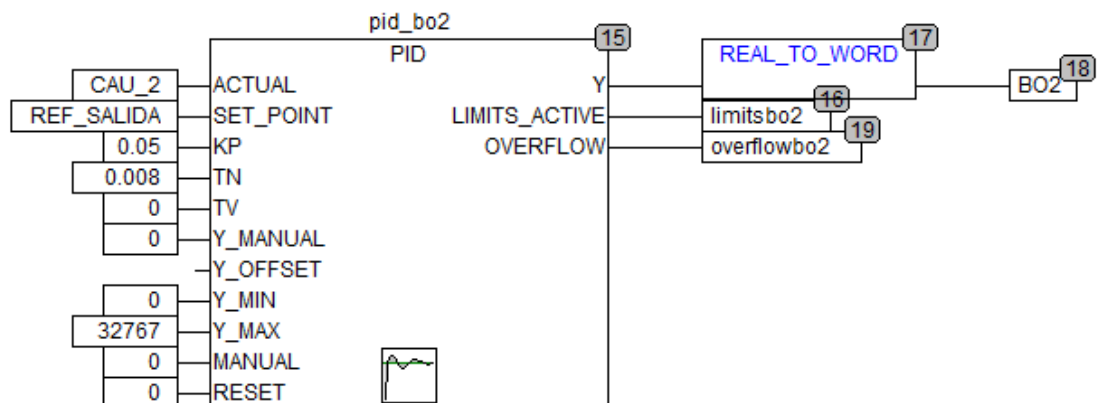
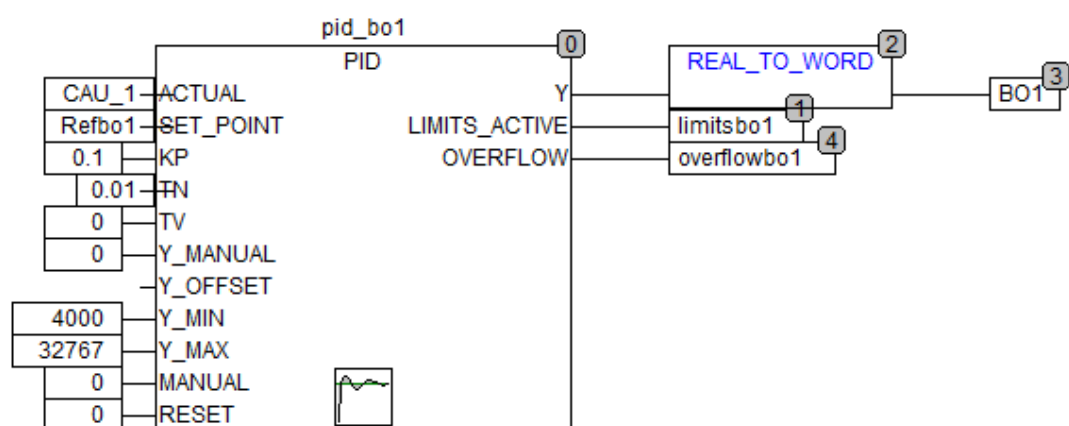
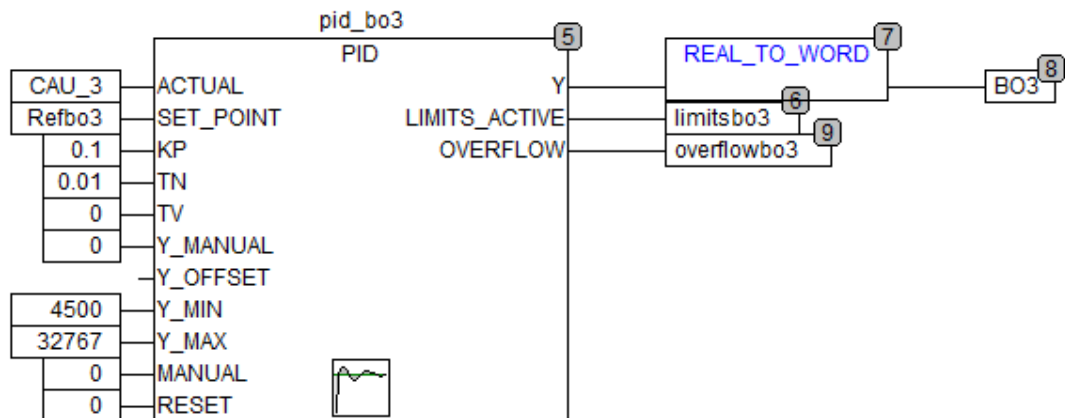
```

Figura 46. Demanda Caudal - Bloque de código 3

6.8.2 PROGRAMA: CONTROL BOMBAS

Esta parte del código está programada en CFC, al contrario que el programa “Demanda Caudal” que estaba en ST. En la figura 47 se presentan los 4 bloques PID que se tienen en este programa, uno por cada bomba de la instalación.

Los parámetros del PID son los mencionados en el final de la sección 6.3, “Estructura de control del caudal”. Las bombas de llenado (1 y 3) reciben como referencia, variables cuyo valor procede de la resolución de las optimizaciones en Matlab. Estas llegan a través de la comunicación MODBUS y son asignadas como referencias a estas bombas. Por otra parte, las bombas de vaciado de los tanques (2 y 4) reciben como referencia la curva de caudal calculada en el programa del PLC explicado anteriormente.



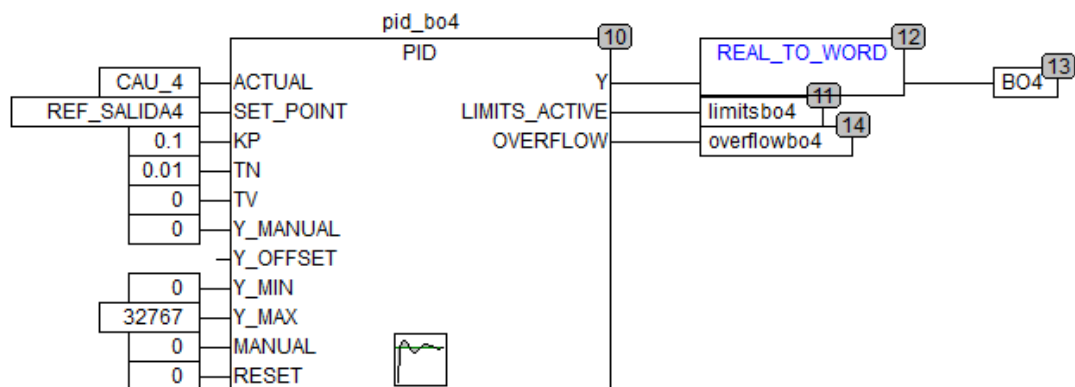


Figura 47. Control Bombas - Bloque de código 1

7 ESTUDIO DE VIABILIDAD

7.1 VIABILIDAD TÉCNICA

Todos los componentes, hidráulicos y eléctricos utilizados en el sistema son elementos comerciales. Además, el algoritmo de optimización se había probado en simulaciones virtuales con anterioridad dando resultados positivos, por lo tanto, el proyecto se considera viable. A posteriori, los experimentos realizados en este proyecto han ratificado esta viabilidad.

7.2 VIABILIDAD ECONÓMICA

Una instalación comercial de laboratorio de características similares a la que se ha ensamblado para este proyecto cuesta en torno a los 30000 euros y el coste final en el que se ha incurrido para llevarlo a cabo apenas supera los 11000 euros. Por lo tanto, resulta económicamente viable.

Por otra parte, en una sola instalación real de tamaño pequeño, con depósitos de 500 m³ y bombas de 200m³/h, se podría alcanzar un ahorro económico anual de 4200 euros, lo que, comparado con el coste del proyecto demuestra su rentabilidad potencial.

8 RESULTADOS

En este apartado se darán algunos datos clave de la implementación de los experimentos no aportados en el apartado de la optimización y se recordarán algunos conceptos.

El problema de optimización final ha sido resuelto con Matlab, usando Yalmip como parser y CBC como solver.

La optimización se resuelve cada 10 periodos, y se realizan experimentos de 2880 periodos de duración (equivalentes a 2 días a tiempo escalado). El periodo de discretización corto es $h = 1.3$ segundos y el largo es $Lh = 30 \cdot 1.3$, es decir, 30 periodos cortos. Con lo que $k_m = 30$ y $k_M = 47$. Si bien, se calculan acciones de control para los 30 primeros periodos en cada optimización, solo las 10 primeras variables de decisión se implementan. Las otras 20, actúan como sistema de garantía, solo se utilizan en caso de que una optimización sea irresoluble y deban aplicarse más elementos del vector de decisión mientras se realiza otra optimización.

Las tarifas utilizadas para cada una de las bombas se pueden observar en la Tabla 3. Tarifas **eléctricas de las bombas** en un desglose en número de periodos (o minutos equivalentes).

Periodos	Bomba 1	Bomba 2
$0 < t < 480$	0.063	0.066
$480 < t < 540$	0.079	0.088
$540 < t < 600$	0.1	0.088
$600 < t < 900$	0.1	0.099
$900 < t < 960$	0.079	0.099
$960 < t < 1440$	0.079	0.088

Tabla 3. Tarifas eléctricas de las bombas

Según estas tarifas los periodos tarifarios caros de la bomba 1 y 2 de 0.1 y 0.099 €/kWh respectivamente son los que se tratarán de evitar en la medida de lo posible a la hora de rellenar los tanques.

Los volúmenes mínimos permitidos para los tanques son $V_{min} = [1; 1]$ y los máximos $V_{max} = [4.5; 4.5]$, en litros. A las 8 de la mañana del primer y segundo día, que se corresponde con los periodos 480 y 1920, se pretende tener los tanques en su nivel máximo permitido (4.5 litros). Los experimentos empiezan a las 12 de la noche con los volúmenes iniciales de los tanques alrededor de 3.8 litros, valor el cual es resultado de la siguiente expresión:

$$V_{inicial} = V_{min} + 0.8 \cdot (V_{max} - V_{min})$$

A continuación, se presentan y analizan distintos casos experimentales y simulados. En primer lugar, se hará una comparativa entre casos experimentales, es decir, utilizando la maqueta hidráulica para realizar los experimentos y casos teóricos en los que las simulaciones son meramente virtuales.

Los distintos casos se caracterizan por las siguientes diferencias entre ellos:

Caso	P11	P21	P22
1	45	60	30
2	60	45	30
3	45	45	30

Tabla 4. Casos de estudio

Donde P11 es la potencia que se consume al bombear desde el pozo 1 al tanque 1, P21 es la potencia que se consume al bombear desde el pozo 2 al tanque 1 y P22 es la potencia que se consume al bombear desde el pozo 2 al tanque 2. Estas potencias están en unidades de KW. Por lo tanto, una prueba experimental y una teórica con el mismo número utilizan las mismas potencias.

En las siguientes gráficas se presenta la evolución de los volúmenes de los tanques, el funcionamiento de las bombas, las curvas de demanda y el estado de las válvulas de las distintas pruebas experimentales y simuladas. Las gráficas del funcionamiento de las bombas tienen superpuestos los periodos tarifarios para observar de forma clara en qué momento se está bombeando.

Experimentos

Las figuras 48, 49 y 50 presentan los resultados de los 3 casos experimentales de estudio en formato gráfico.

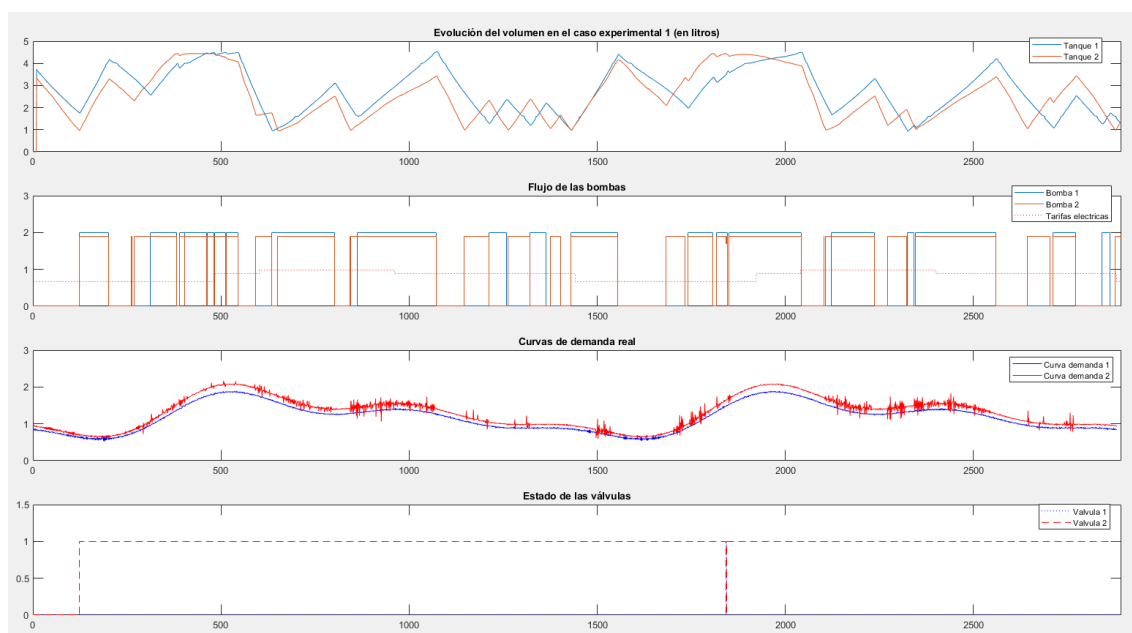


Figura 48. Resultados del caso experimental 1

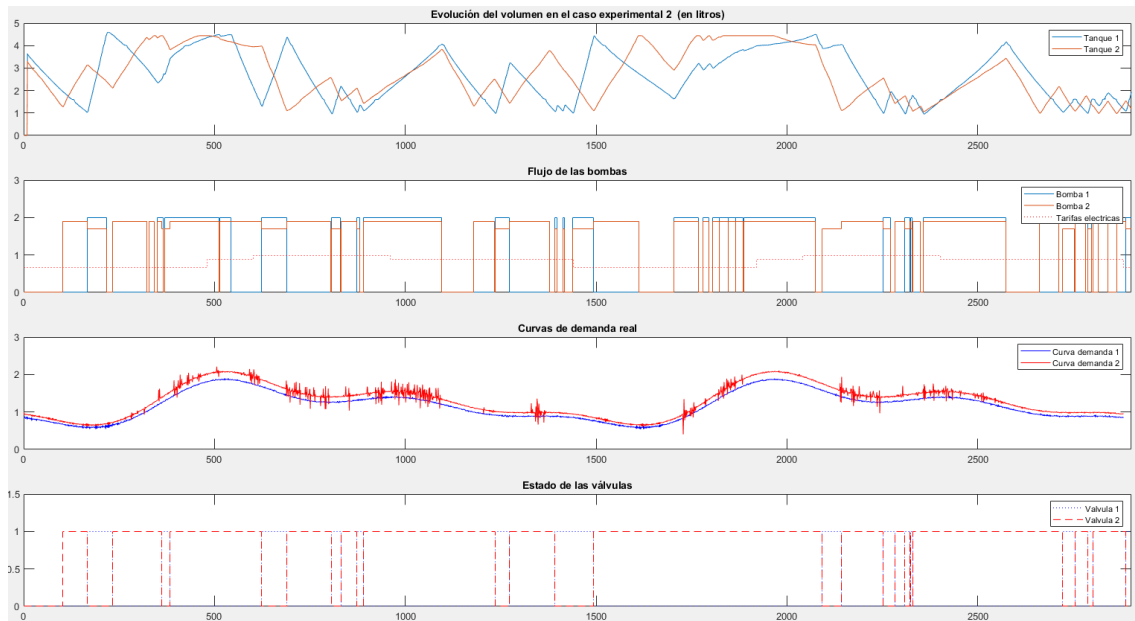


Figura 49. Resultados del caso experimental 2

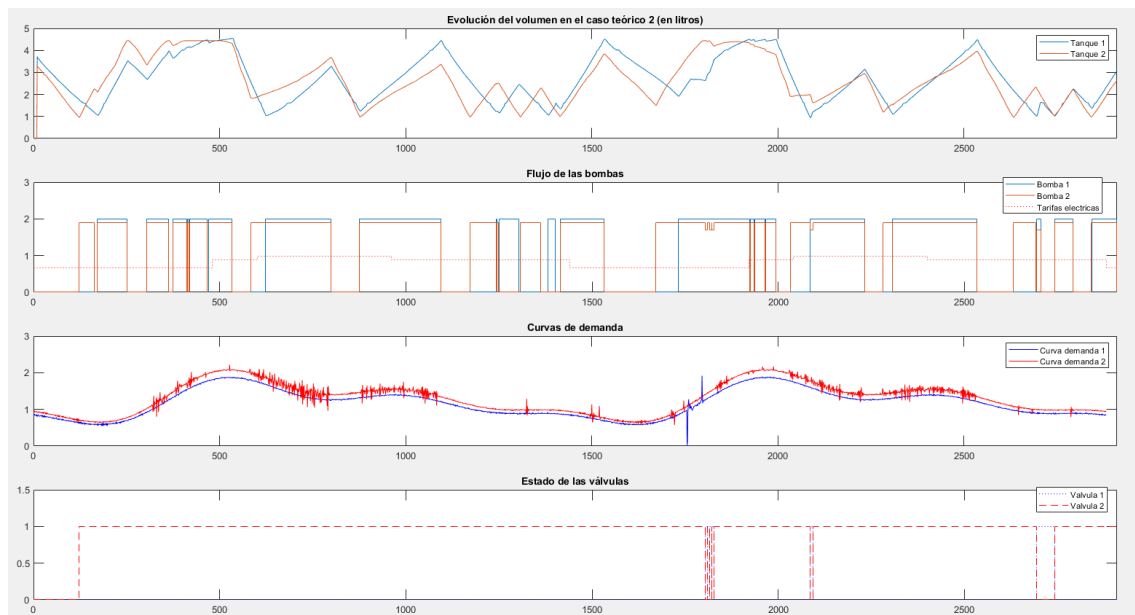


Figura 50. Resultados del caso experimental 3

Casos		%ON - B1	%ON - B2	Vol. 1 - 8am	Vol. 2 - 8am	Jtot(€)	ConnTot	Infeas
Experimentales	C1	55,7	66,0	4,5	4,4	176,1	69,0	9
	C2	44,2	79,6	4,3	4,4	205,3	86,0	27
	C3	55,8	67,5	4,4	4,4	179,4	70,0	4

Tabla 5. Comparativa de las pruebas experimentales

En la Tabla 5. **Comparativa de las pruebas experimentales** podemos observar una comparativa entre las pruebas experimentales. En esta se recogen datos como el

porcentaje de tiempo durante el que se han usado las distintas bombas, el volumen medio alcanzado en cada tanque a las 8 de la mañana, el índice de coste (J_{tot}), el número de conmutaciones totales ($Conmtot$) y el número de optimizaciones irresolubles ($Infeas$).

De los porcentajes de tiempo de uso de las bombas se puede observar el uso de una bomba u otra en función de las potencias que se le asignan a cada una. En los casos 1 y 3 se obtienen resultados muy similares puesto que las potencias de bombear al tanque 1 desde el pozo 2 son iguales o superiores a la potencia de bombear al tanque 1 desde su propia bomba. Por lo tanto, la optimización se decanta por utilizar mayoritariamente la bomba 1 en exclusiva para llenar este tanque en ambos casos.

Por otra parte, el índice de coste del caso 2 es superior al del caso 1 y 3. Esto es así ya que en el caso 2, como se observa en la Figura 49, las bombas rara vez se encuentran paradas llegando a estar la bomba 2 casi un 80% del tiempo del experimento encendida. Además de que durante los periodos tarifarios caros siempre hay alguna bomba activa. Esto es consecuencia de que en el caso 2 la bomba 1 pasa a tener una potencia de 60 kW al bombear a su propio tanque y en cambio rellenarlo desde la bomba 2 consume solo 45, por lo tanto, se incentiva el uso de la bomba 2 para rellenar el tanque 1. Sin embargo, cuando la bomba 2 llena el tanque 1, esta solo puede trasegar un flujo de 1.7 litros por minuto en comparación con los 2 litros por minuto de la bomba 1. Esto hace que sea más difícil llenar el tanque 1 a un bajo precio y reduce la disponibilidad de la bomba 2 que ahora se usa para los dos tanques. El resultado final es un problema de optimización más complejo para la rápida dinámica del experimento, que conlleva mayor número de errores ($infeas$), conmutaciones y finalmente un mayor coste monetario.

Simulaciones

Las figuras 51, 52 y 53 presentan los resultados de los 3 casos simulados de estudio en formato gráfico.

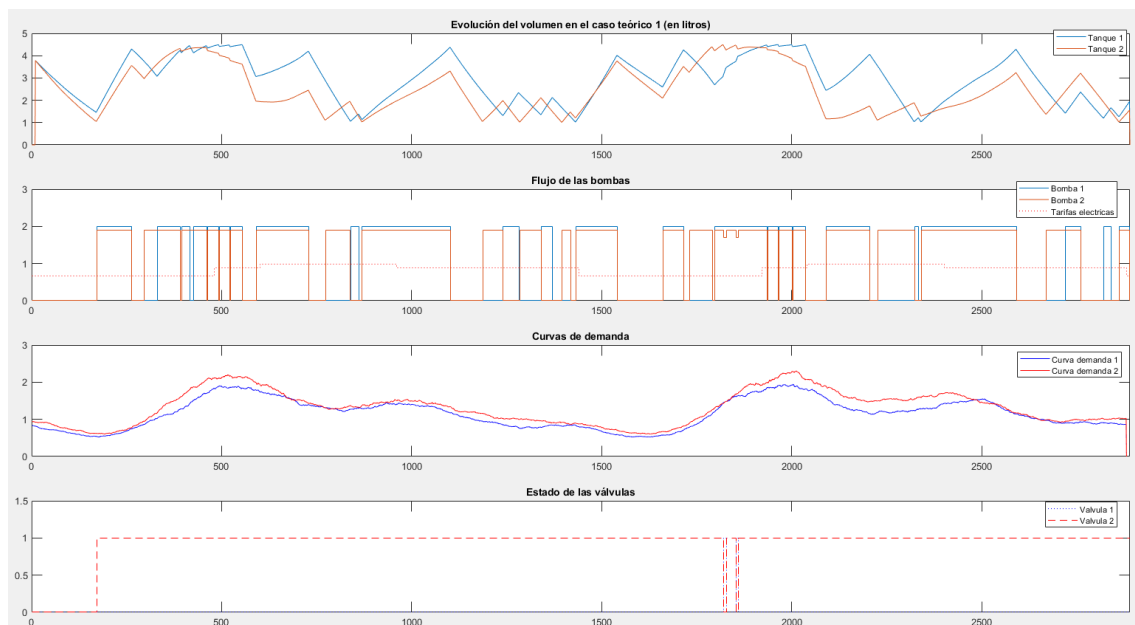


Figura 51. Resultados del caso teórico 1

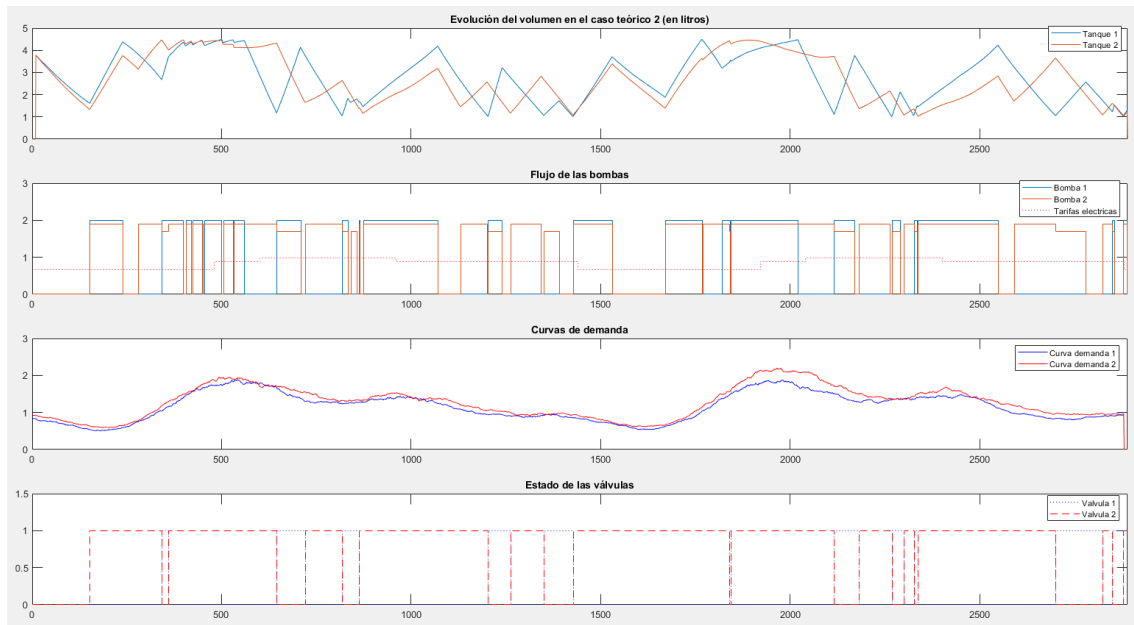


Figura 52. Resultados del caso teórico 2

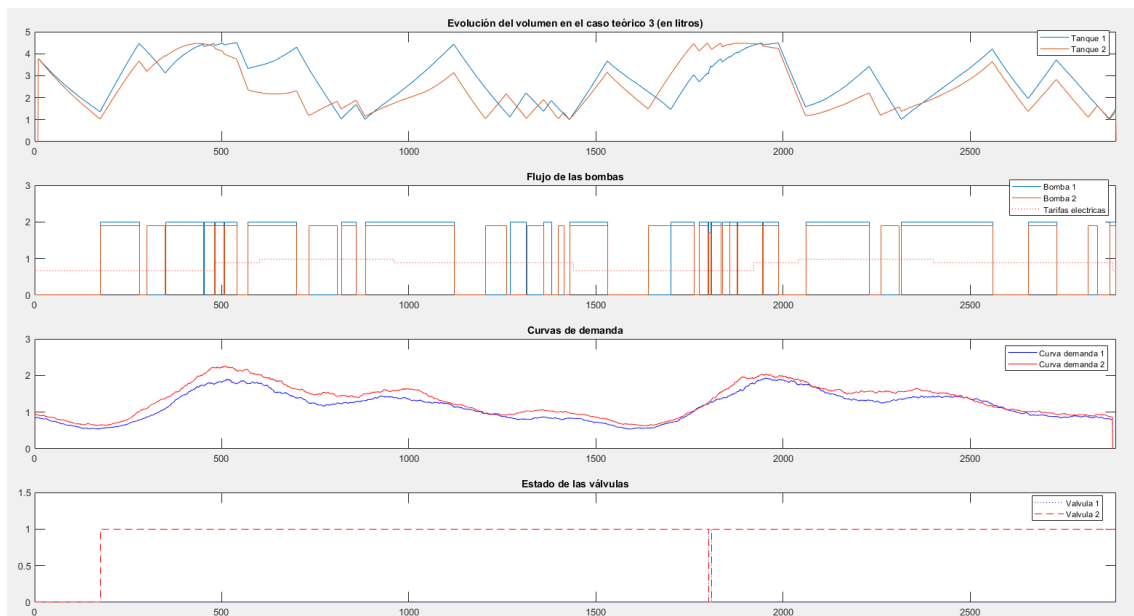


Figura 53. Resultados del caso teórico 3

Casos		%ON - B1	%ON - B2	Vol. 1 - 8am	Vol. 2 - 8am	Jtot(€)	ConnTot	Infeas
Experimentales	C1	55,7	66,0	4,5	4,4	176,1	69	9
	C2	44,2	79,6	4,3	4,4	205,3	86	27
	C3	55,8	67,5	4,4	4,4	179,4	70	4
Teóricos	C1	55,9	67,2	4,5	4,5	178,9	68	7
	C2	45,0	77,2	4,4	4,5	204,7	74	7
	C3	56,1	67,0	4,4	4,3	178,5	55	6

Tabla 6. Comparativa de las pruebas teóricas

La tabla anterior nos presenta una comparación entre las pruebas con la maqueta hidráulica y las simulaciones por ordenador para unas mismas potencias de las bombas.

Así pues, se puede observar que las diferencias entre un caso experimental y teórico del mismo número son mínimas, habiendo únicamente una cierta diferencia en el número de conmutaciones. Los casos experimentales presentan un mayor número de conmutaciones que los teóricos de forma general, siendo esto más acentuado en los casos 2 y 3. Aun con esto se puede afirmar que los resultados tanto en las pruebas experimentales como teóricas presentan una gran similitud y que, por lo tanto, las pruebas experimentales confirman los resultados de las optimizaciones en las simulaciones por ordenador. El cual es uno de los objetivos principales de este proyecto.

Control todo-nada simple

Para terminar, también se ha probado en la maqueta un sistema simple de control todo-nada, para compararlo con el control óptimo. En este caso se supone que cada bomba solo llena su depósito, y los umbrales de conmutación los marcan los niveles máximo y mínimo de 4.5 y 1 litro. De esta forma se pretende comparar los costes económicos en los que repercute el uso de un control todo-nada en contraposición al uso de la metodología propuesta en este trabajo.

La Figura 54. **Resultados de la prueba Relé** tiene representados en la gráfica superior la evolución del volumen de un tanque aplicando el control relé al sistema y en la inferior el uso de la bomba y los periodos tarifarios.

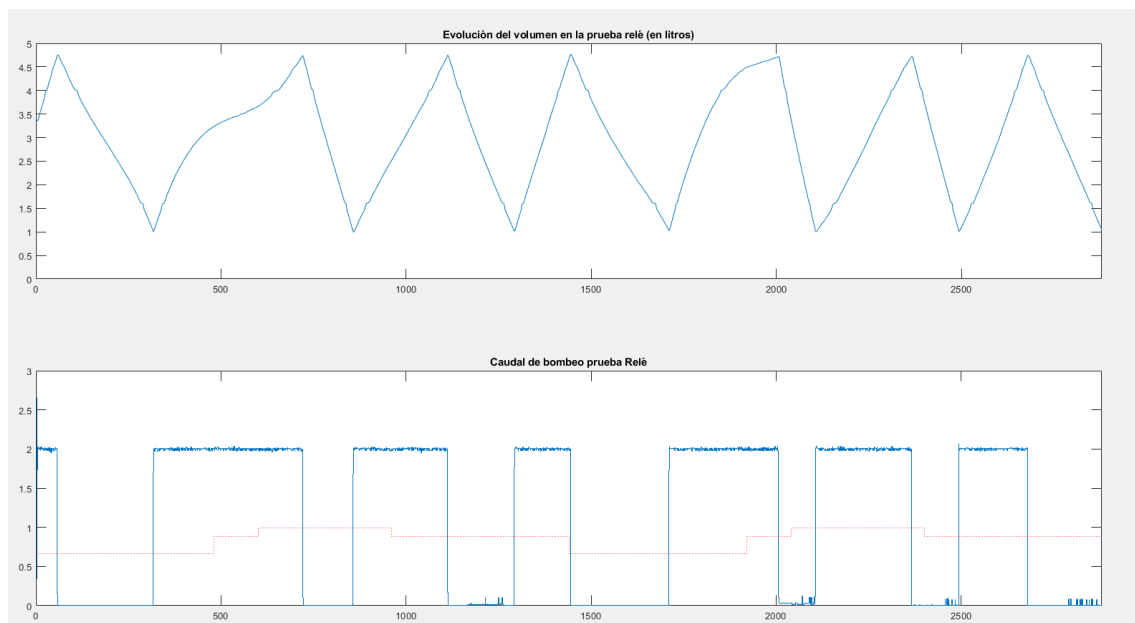


Figura 54. Resultados de la prueba Relé

Casos		%ON - B1	%ON - B2	Vol. 1 - 8am	Vol. 2 - 8am	Jtot(€)	ConnTot
Prueba Relé	C1	55,8681	55,7624	-	-	198,8655	14,0
Experimentales	C1	-	-	-	-	175,1001	-

Tabla 7. Comparativa entre prueba relé y casos experimentales

Las bombas se encuentran bombeando algo más de la mitad del tiempo durante este tipo de control y, como se puede observar en la Figura 54. **Resultados de la prueba Relé**, la conmutación de la bomba es independiente de la tarifa, mientras que con la optimización se intenta reducir lo máximo posible el tiempo de bombeo en los periodos tarifarios más caros.

A partir de los datos de la tabla 7 se puede determinar que el índice de coste obtenido en el control relé, únicamente teniendo en cuenta la potencia consumida, es superior en un 13% al índice de coste al aplicar el algoritmo de optimización. Con lo que, si se tiene en cuenta que el índice de coste es de 2 días virtuales de experimento, el ahorro de aplicar este control al final de un mes de operación sería de unos 356 €.

9 CONCLUSIÓN Y TRABAJOS FUTUROS

9.1 CONCLUSIÓN

En este proyecto se ha desarrollado el sistema hidráulico y eléctrico de la planta piloto, se han ajustado las características de los componentes para que la escala, tanto a nivel de caudal como de tiempo, permitan emular un sistema de bombeo real a pequeña escala. También se ha programado el funcionamiento de los distintos elementos de la planta, la estructura de control PID del caudal, el sistema de comunicación MODBUS y la optimización de la gestión de las bombas y las válvulas.

De los resultados se puede derivar que la implementación práctica del algoritmo de optimización utilizado en este proyecto verifica los resultados obtenidos teóricamente y que supone un ahorro económico sobre un supuesto control tipo relé.

Al tener que utilizar un periodo de temporización mayor al inicialmente planificado los tiempos simulados de llenado de los tanques han sido ligeramente mayores de lo esperado, es decir, que las bombas han tenido que trabajar durante más porcentaje de tiempo y no se ha podido reducir más los costes. Si se reduce la relación entre el caudal de salida y la capacidad de bombeo, el porcentaje de tiempo en el que las bombas funcionan se reduciría, por lo que la optimización permitiría reducir el tiempo de bombeo en instantes más caros, y el ahorro respecto al control todo-nada sería mayor.

9.2 TRABAJOS FUTUROS

En futuros trabajos se podría intentar permitir un mayor caudal de llenado de los tanques en comparación con el caudal de salida para poder observar de forma más clara la gestión de las bombas y la reducción de costes debidas al algoritmo de optimización.

Por otra parte, este algoritmo podría ser implementado en una instalación de bombeo real. También se propone utilizar un cliente y servidor OPC para realizar las comunicaciones entre la optimización y el PLC, ya que de esta forma se podría integrar con el servidor OPC del que habitualmente muchas empresas ya disponen.

10 BIBLIOGRAFÍA

- Sanchis, R. (2017). Minimum cost control of pumping systems with scheduled electric tariffs. *6th International Conference on Advanced Technology & Sciences(ICAT'Riga)*, (págs. 221-228). Riga, Latvia.
- Manual Wago Modular I/O-System ETHERNET TCP/IP 750-841
<http://www.prosoft.ru/cms/f/436944.pdf>
- Wikipedia. www.wikipedia.org
- User Manual for PLC Programming with CoDeSys 2.3
http://www.ee.pw.edu.pl/~purap/PLC/manuals/m07590333_00000000_1en.pdf
- Matlab Documentation. <https://es.mathworks.com/help/>

ANEXOS

1 LISTADO DEL CÓDIGO DE MATLAB

1.1 PROGRAMA: INICIALIZA

16/10/19 17:25 D:\JONA\Trabajos Johnata\...\inicializa.m 1 of 2

```

yalmip clear

% Matrices que definen los pozos activos
%y la potencia de los pozos en función de las combinaciones (Delta*P, Delta*P1):

P11 = 15; %Potencia de la bomba 1 al llenar al deposito 1
P21 = 20; %Potencia de la bomba 2 al llenar al deposito 1
P22 = 10; %El depósito 2 esta más bajo; Lo suponemos

P = [0 P11 0 0 P11 P11; 0 0 P21 P22 P22 P21];
Sp = [0 1 0 0 1 1; 0 0 1 1 1 1];

% Matriz que define la válvulas activas en función de la combinaciones (Delta*W)
X=3;
W = [X X 1 0 0 1; X X 0 1 1 0];

% Vectores que definen los caudales de llenado de los depósitos
%en función de las combinaciones (Delta*q)
Q11 = 2;
Q22 = 1.9;
Q21 = 1.7; %Estas serán las referencias de las bombas

q_1 = [0 Q11 Q21 0 Q11 Q11+Q21];
q_2 = [0 0 0 Q22 Q22 0];

FT = [ q_1; q_2]; %CAUDAL QUE LE LLEGA A CADA DEPÓSITO EN CADA UNA DE LAS 6
COMBINACIONES

Fp = [0 Q11 0 0 Q11 Q11; 0 0 Q21 Q22 Q22 Q21];

Nc=6; %n° COMBINACIONES VALVULAS, BOMBAS TANQUES
Nv=2;
Np=2;
Nt=2;

T_tiempoReal = 1.3; % segundos

%Volúmenes
Vmin=[1;1];
Vmax=[4.5;4.5];
Vinicial=Vmin+0.8*[1;1].*(Vmax-Vmin); %Esto es para la simulación, la situación
inicial en la que supondremos o estaran los tanques

a1=0.95;
a2=0.95;
a3=0.95;

combinaciones=6;

Jnumconm=true;
if exist('alfa')==0 alfa=0.05; end
Resnumconm=false;
if (exist('numconmmmax')==0) numconmmmax=32; end
binarias=false;
solver='cbc';

```

16/10/19 17:25 D:\JONA\Trabajos Johnata\...\inicializa.m 2 of 2

```
% Definición de caudales de salida de los depósitos (consumidos por los usuarios):
tolgap=0.05; % tolerancia admisible en la optimización
tolgapmax=0.4; % tolerancia admisible si pasa el tiempo máximo de ejecución
maxtiempo=180;

% Definición de tarifas de los pozos (euros/kWh cada minuto)

T1f=[0.063*ones(1,480),0.079*ones(1,540-480),0.1*ones(1,900-540),0.079*ones(1,1440-900)]; %Tarifas de cada bomba en euros por kilovatio hora minuto a minuto; 480 = 8 de la mañana Tarifas 3.0 y 3.1
T2f=[0.066*ones(1,480),0.088*ones(1,600-480),0.099*ones(1,960-600),0.088*ones(1,1440-960)];

Deltaaplicada=zeros(Nc*tsimtotal,1);
Volmedido=zeros(2,tsimtotal);
Fsalidaestimada=ones(2,1441)*1.2;
Qmedido=zeros(2,tsimtotal);
Qbombeo=zeros(2,tsimtotal);
Valvula=zeros(2,tsimtotal);
a0 = 97.64; a1 = -36.18; b1 = 1.699; a2 = -4.421; b2 = -17.57; a3 = 14.34; b3 = 4.85;
w = 0.2618;
a=[a1;a2;a3];b=[b1;b2;b3];
nf=3; %Simulacion caudal de salida
FRACAUDAL=[0.0133*0.9;0.0133];
for iti=1:ndias*1441
    Fbase = a0 + cos((iti+tni)/60*w*(1:nf))*a + sin((iti+tni)/60*w*(1:nf))*b;
    Fsalidaestimada(:,iti)=[Fbase;Fbase].*FRACAUDAL;
end
```

1.2 PROGRAMA: LLAMA AL OPTI

16/10/19 17:25 D:\JONA\Trabajos Johna...\llama al opti.m 1 of 6

```
clear all;
global bandera;
global direcc_tiempoMatlab;
bandera = 1;

global direcc_nivel1;
global direcc_nivel2;
global Volmedido;
global Qmedido;
global Qbombeo;
global direcc_bo1;
global direcc_bo3;
global Valvula;
global Deltaaplicada;
global Fp;
global Nc;
global Fsalidaestimada;
global perexpoQ;
global direcc_caudal2;
global direcc_caudal4;
global direcc_ev4;
global direcc_ev3;
global W;
global Ferror;

%Direcciones de memoria

direcc_periodo = 12307;
direcc_caudal_salida = 12308;
direcc_caso = 12309;
direcc_tiempoMatlab = 12310;

direcc_bo1 = 12289;
direcc_bo2 = 12290;
direcc_bo3 = 12291;
direcc_bo4 = 12292;

direcc_caudal1 = 12293;
direcc_caudal2 = 12294;
direcc_caudal3 = 12295;
direcc_caudal4 = 12296;

direcc_nivel1 = 12297;
direcc_nivel2 = 12298;

direcc_ev1 = 12299;
direcc_ev2 = 12300;
direcc_ev3 = 12301;
direcc_ev4 = 12302;

direcc_lect_bo1 = 12303;
direcc_lect_bo2 = 12304;
direcc_lect_bo3 = 12305;
direcc_lect_bo4 = 12306;
```

16/10/19 17:25 D:\JONA\Trabajos Johna...\llama al opti.m 2 of 6

```

global t; %Para que use el t que se incrementa en el timer

ndias=2; % 2 días
infeas=[];
infeas2=[];
tinicial=10;
tini=tinicial;
tsimtotal=ndias*1440+tini;
inicializa;
conmtot=0;
Jtot=0;
Vmaxobj=Vmin+0.95*(Vmax-Vmin);
tVmax=480; %8 DE LA MAÑANA, ES CUANDO QUEREMOS QUE LOS DEPÓSITOS ESTEN LLENOS DEBIDO A✓
QUE ES CUANDO SE TERMINA EL PERIODO BARATO
t=tini;
Vini=Vinicial;
Deltaini=zeros(Nc,1);
periodooptim=10;
periodo=1.3;
perexpoQ=60; %Factor de corrección exponencial en minutos, durante esto minutos se va✓
corrigiendo la predicción de caudal
km=30;
kul=20;
ku2=60;
ku=kul;
L=30/periodo;
KM=(1440-km*periodo)/L/periodo+km;
numconmmax=48;
excV=0.1; %Factores de ponderación de la optimización
alfa=0.1;%Factores de ponderación de la optimización
alfaV=0.001;%Factores de ponderación de la optimización
soloJ=false;
JVmaxobj=false; %Que no incluya el volumen máximo objetivo en el índice de coste, sino✓
que este por defecto en las restricciones, si da tiempo a que se cumpla las✓
restricciones de las 8 de la mañana, si no se puede cumplir se pone en el índice de✓
coste
Jnumconm=true; %Le suma un factor al índice de coste para penalizar el numero excesivo✓
de conmutaciones
resfinal=true; % Restricción de que acaben los niveles donde habían empezado (24✓
horas)
ResVmaxobj=true; %Volumen máximo objetivo a las 8 de la mañana como restricción
Resnumconm=false; %Numero de conmutaciones como restricción
if soloJ==true %pasa el ncom y Vmax Objetivo al índice de coste
    JVmaxobj=true;
    ResVmaxobj=false;
end
optimiza_bombeo_futuro
if sol.problem==1
    infeas=[infeas,1];
    JVmaxobj=true;
    ResVmaxobj=false;
    optimiza_bombeo_futuro

    if sol.problem==1
        infeas2 = [infeas2,1];
    
```

16/10/19 17:25 D:\JONA\Trabajos Johna...\llama al opti.m 3 of 6

```

        else
            infeas2=[infeas2,0];
        end

    else
        infeas=[infeas,0];
    end
    JVmaxobj=false;
    ResVmaxobj=true;
    if soloJ==true
        JVmaxobj=true;
        ResVmaxobj=false;
    end
    for ik=1:periodooptim
        Deltaaplicada(t*Nc+Nc*(ik-1)+1:t*Nc+Nc*ik)=value(Deltab((ceil(ik/periodo)-1)*Nc+1:
Nc*ceil(ik/periodo)));
    end

Deltaini=Deltaaplicada(t*Nc+Nc*periodooptim-Nc+1:t*Nc+Nc*periodooptim); %Ultima delta
aplicada antes de la siguiente optimización, si no se pone al inicio de cada
optimización se haría una conmutación, lo cual a lo largo el día serían demasiadas
conmtot=conmtot+conmbo; %Acumulación del numero de conmutaciones que se van a aplicar
(en las diez primeras deltas)
Jtot=Jtot+Jo; %Jo valor del indice de coste que se tiene durante los 10 primeros
deltas y su acumulado durante la simulación Jtot

periodo_codesys = 1.3;

tv = timer('StartDelay', 0, 'Period', periodo_codesys, 'ExecutionMode', 'fixedRate');
tv.TimerFcn = @(~,thisEvent)Timer_Lab;

%ENVIAR ORDEN DE QUE ACTIVE O DESACTIVE VÁLVULAS Y ESCRIBIR LOS CAUDALES DE
%REFERENCIA DE LAS BOMBAS DE LLENADO
global comienzo;
m = modbus('tcpip', '192.168.1.3', 502);
    Nivel1_PLC = read(m, 'holdingregs', direcc_nivel1,1);
    Nivel1_Litros = Nivel1_PLC*(4.867/32761);
    Nivel2_PLC = read(m, 'holdingregs', direcc_nivel2,1);
    Nivel2_Litros = Nivel2_PLC*(4.44/32761);
Volmedido(:,t)= [Nivel1_Litros; Nivel2_Litros]; %MEDICION DE LOS NIVLES DE LOS
DEPOSITOS
    QSalida1_PLC = read(m, 'holdingregs', direcc_caudal2,1);
    QSalida2_PLC = read(m, 'holdingregs', direcc_caudal4,1);
    QSalida1 = QSalida1_PLC*(8/16383);
    QSalida2 = QSalida2_PLC*(8/16383); %litros por minuto
Qmedido(:,t)= [QSalida1;QSalida2]; %medicion de los caudales de salida
Qbombero(:,t)= Fp*Deltaaplicada((t-1)*Nc+1:t*Nc); %Caudal que bombeamos
    Qbombero1 = Qbombero(1,t);
    Qbombero2 = Qbombero(2,t);
    Qbombero1_PLC = round(Qbombero1*(16383/8));
    Qbombero2_PLC = round(Qbombero2*(16383/8));
    write(m, 'holdingregs', direcc_bo1, Qbombero1_PLC);
    write(m, 'holdingregs', direcc_bo3, Qbombero2_PLC);

```

16/10/19 17:25 D:\JONA\Trabajos Johna...\llama al opti.m 4 of 6

```

Valvula(:,t)=W*Deltaaplicada((t-1)*Nc+1:t*Nc); %Situación de las válvulas
for ij=1:2
    if Valvula(ij,t)==3
        Valvula(ij,t)=Valvula(ij,t-1); %Si el resultado de la optimización da que la
        válvula este en 3, no importa su estado, se deja como estaba
    end
end
valvula4 = Valvula(1,t);
valvula3 = Valvula(2,t);
write(m,'holdingregs' , direcc_ev4, valvula4);
write(m,'holdingregs' , direcc_ev3, valvula3);

Fsalidaestimada(:,1:1440)=Fsalidaestimada(:,2:1441); %Registro el nuevo valor estimado
del caudal de salida, corriendo en 1 el vector.
Error=Qmedido(:,t)-Fsalidaestimada(:,1); %Comparo el caudal medido de salida con el
caudal estimado y obtengo el error de mi estimación
Fsalidaestimada(:,1)=Qmedido(:,t); %Corrijo el valor actual con el que acabo de medir
Fsalidaestimada(:,1441)=Qmedido(:,t); %Corrijo tambien el de dentro de 24 horas puesto
que supongo que sigue un patron periodico
for i=2:perexpoQ*4
    Fsalidaestimada(:,i)=Fsalidaestimada(:,i)+Error*exp(-i/perexpoQ); %Corrijo
sumando una fracción del error cada vez en un tiempo perexpoQ=60 *4 de los siguientes
valores perexpoQ=60 *4 (4 horas)
end

while (t<tsimtotal)
    if(bandera == 1)

if mod(t,periodooptim)==0 %numero de minutos es multiplo de 10 se ejecuta la
optimización
    t_inicial=t;
    Vini=Volmedido(:,t) %Definimos el volumen inicial, que es el que acabo de medir
    tini=mod(t,1440) %numero de minuto dentro del dia en el que se inicia la
simulación (puesto que t se aumenta de forma infinita)
    ku=ku1;
    optimiza_bombeo_futuro
    if sol.problem==1
        infeas=[infeas,1];
        ku=ku2;
        JVmaxobj=true;
        resfinal=false;
        ResVmaxobj=false;
        optimiza_bombeo_futuro

        if sol.problem==1
            infeas2 = [infeas2,1];
        else
            infeas2=[infeas2,0];
        end
    end

    else
        infeas=[infeas,0];
    end
end

```

16/10/19 17:25 D:\JONA\Trabajos Johna...\llama al opti.m 5 of 6

```

JVmaxobj=false;
resfinal=true;
ResVmaxobj=true;
if soloJ==true
    JVmaxobj=true;
    ResVmaxobj=false;
end

if (sol.problem==0)
for ik=(t-t_inicial+1):periodooptim*3
    Deltaaplicada(t_inicial*Nc+Nc*(ik-1)+1:t_inicial*Nc+Nc*ik)=value(DeltaTab((ceil(
(ik/periodo)-1)*Nc+1:Nc*ceil(ik/periodo))));
end
end
conmtot=conmtot+sum(abs(Ybo*[Deltaini;Deltaaplicada(t*Nc+1:t*Nc+Nc*periodooptim-
Nc)]))

Deltaini=Deltaaplicada(t_inicial*Nc+Nc*periodooptim-Nc+1:
t_inicial*Nc+Nc*periodooptim);

Jtot=Jtot+I*Pg(:,1:periodooptim/periodo*Nc)*Deltaaplicada(t_inicial*Nc+1:
t_inicial*Nc+Nc*periodooptim);

end
% Ejecución Timer
bandera = 0;
if isempty(comienzo)
    start(tv)
    write(m,'holdingregs' , direcc_periodo, periodo_codesys*10);
    comienzo = 1;

end

end

end
end

conmtot=conmtot+conmbo;
Jtot=Jtot+Jo;

figure(1)
subplot(4,1,1)
hold off

plot(((Volmedido-Vmin*ones(1,length(Volmedido)))/(Vmax-Vmin)*ones(1,length(
(Volmedido)))*100)');
axis([0,length(Volmedido),-10 110]);
hold on

title('Volumes (in %)');
subplot(4,1,2)
hold off
stairs([Qbombeo(1,:)';Qbombeo(1,length(Qbombeo))']');
hold on
Tlfl=[];

```

16/10/19 17:25 D:\JONA\Trabajos Johna...\llama al opti.m 6 of 6

```

for ik=1:ndias
    T1f1=[T1f1,T1f];
end
T1f1=[T1f1,T1f(1:480)];
stairs(10*[T1f1], 'r:');

axis([0,length(Qbombeo),0 3]);
title('Flow of pump 1');
subplot(4,1,3)
hold off
stairs([Qbombeo(2,:);Qbombeo(2,length(Qbombeo))]);
hold on
T2f1=[];
for ik=1:ndias
    T2f1=[T2f1,T2f];
end
T2f1=[T2f1,T2f(1:480)];
stairs(10*[T2f1], 'r:');

axis([0,length(Qbombeo),0 3]);
title('Flow of pump 2');

subplot(4,1,4)
hold off
stairs([Valvula(1,:);Valvula(1,length(Valvula))], 'b:');
hold on
stairs([Valvula(2,:);Valvula(2,length(Valvula))], 'r--');
axis([0,length(Valvula),0 1.5]);
title('Valves');
figure(3)
hold off;
plot(Qmedido(:,tinicial:tinicial+1440*ndias), 'b');
hold on;
plot([Fsalidaestimada], 'r');

```


1.3 PROGRAMA: OPTIMIZA BOMBEO FUTURO

16/10/19 17:25 D:\JONA\Traba...\optimiza bombeo futuro.m 1 of 5

```

yalmip clear
%clear all;
%Variables a definir antes de llamar a este script:
% tini = minuto actual dentro del día cuando se ejecuta la optimización
% Fsalidaestimado=matriz de 2x1440 con los valores estimados de caudal de
% salida para las próximas 24 horas
% Vini= vector con tres elementos con los volúmenes iniciales
% tVmax= tiempo en minutos del día donde se requiere volúmenes máximos
% Vmaxobj= vector con los volúmenes objetivos en el instante anterior
% periodo = periodo de muestreo en segundos en la optimización cercana (1.3)
% L= número de periodos pequeños para definir el periodo en la optimización
% lejana (30)
% Horizonte = número de minutos del horizonte de optimización (1440)
% km = número de periodos de la parte de optimización entera (30)
% KM = número de periodos total (30+47)
% KM=(Horizonte-km*periodo)/L;
% Jnumconm = booleano para indicar que se añade al índice de coste el número de
conmutaciones (true);
% alfa= coeficiente de ponderación del número de conmutaciones en el índice de coste
% Resnumconm= booleano para indicar que se añade como restricción el número de
conmutaciones
% en la parte de optimización entera (false);
% numconmmax= número de conmutaciones máximo en la parte de optimización próxima
entera (2)
% JVmaxobj = booleano para indicar que se añade al índice de coste el error
% respecto Vmaxobj
% alfaV = coeficiente de ponderación del error respecto Vmaxobj en el índice de coste
% ResVmaxobj = booleano para indicar que se incluye como restricción
% alcanzar el valor Vmaxobj en el instante tVmax (false)

if exist('alfa')==0 alfa=0.05; end
Resnumconm=false;
if (exist('numconmmax')==0) numconmmax=32; end

if (Vini<Vmax)
    advmax=0;
else
    advmax=0.95;
end
if (Vini>Vmin)
    advmin=0;
else
    advmin=0.95;
end

% Definición de tarifas de los pozos (euros/kWh cada minuto)
T1=[T1f(tini+1:length(T1f)),T1f(1:tini)]; %Corres el vector para que sea desde este
instante hasta dentro de 24 horas
T2=[T2f(tini+1:length(T2f)),T2f(1:tini)];

YYb=eye(km*Nc);
YMb=YYb(Nc+1:km*Nc,:);
Ymb=YYb(1:(km-1)*Nc,:);
Yb=YMb-Ymb;

```

16/10/19 17:25 D:\JONA\Traba...\optimiza bombeo futuro.m 2 of 5

```

YYbo=eye(periodooptim/periodo*Nc);
YMbo=YYbo(Nc+1:periodooptim/periodo*Nc,:);
Ymbo=YYbo(1:(periodooptim/periodo-1)*Nc,:);
Ybo=YMbo-Ymbo;
YY=eye(KM*Nc);
YM=YY(Nc+1:KM*Nc,:);
Ym=YY(1:(KM-1)*Nc,:);
Y=YM-Ym;
T=[];
Pg=[];
I=[];
F=[];
Ff=[];
Fo=[];
Foreal=[];
S=[];
Sf=[];
for i=1:km
    T=[T,periodo/60*T1(1,(i-1)*periodo+1),periodo/60*T2(1,(i-1)*periodo+1)];
    Pg=blkdiag(Pg,P); %Pone en la diagonal principal la matriz P sucesivamente
    I=blkdiag(I,ones(1,Nc)); %Para poner la restricción de que las deltas tienen que
valer 1 al sumarse
    Ff=[Ff,FT];
    F=[F;[Ff,zeros(Nt,KM*Nc-length(Ff))]];

    S=blkdiag(S,Sp);
    if (i==1)

        Fo=Fsalidaestimada(:,1);
    else
        Fo=[Fo,sum(Fsalidaestimada(:,1:i*periodo)')'/periodo];
    end
end
for i=km+1:KM
    T=[T,periodo*L/60*T1(1,(km-1)*periodo+(i-km)*periodo*L+1),periodo*L/60*T2(1,(km-1)
*periodo+(i-km)*periodo*L+1)];
    Pg=blkdiag(Pg,P);
    I=blkdiag(I,ones(1,Nc));
    Ff=[Ff,FT*L];
    F=[F;[Ff,zeros(Nt,KM*Nc-length(Ff))]];
    S=blkdiag(S,Sp);
    Fo=[Fo,sum(Fsalidaestimada(:,1:km*periodo)')'/periodo+sum(Fsalidaestimada(:,
km*periodo+1:km*periodo+(i-km)*L*periodo)')'/periodo];
end

Vinf=zeros(KM,Nt);
Vsup=zeros(KM,Nt);

Deltab=binvar(km*combinaciones,1); %Variables binarias de optimización
Deltaf=sdpvar((KM-km)*combinaciones,1); %Variables reales de optimización
Delta=[Deltab;Deltaf];
J=T*Pg*Delta;
if (Jnumconm==true)
    J=J+alfa*sum(abs(Yb*[Deltaini;Deltab(1:length(Deltab)-Nc)])); %Resta una delta

```

16/10/19 17:25 D:\JONA\Traba...\optimiza bombeo futuro.m 3 of 5

```

entre la anterior para medir las conmutaciones Yb
end
if (JVmaxobj==true) || (ResVmaxobj==true)
    dtVmaxtini=tVmax-tini;
    if dtVmaxtini<=0 dtVmaxtini=dtVmaxtini+1440; end
    if dtVmaxtini<=km*periodo
        indVmax=round(dtVmaxtini/periodo);
    else
        indVmax=km+round((dtVmaxtini/periodo-km)/L)
    end
    if indVmax==0 indVmax=KM; end
end
if (JVmaxobj==true)
    J=J+alfaV*[1 1]*(60*(Vmaxobj-Vini)/periodo+Fo(:,indVmax)-F((indVmax-1)*Nt+1:
(indVmax-1)*Nt+Nt,:)*Delta)./(Vmax-Vmin));
end
Res=[0<=Delta<=1];

if (sum(Vini>Vmax)==0)&&(sum(Vini<Vmin)==0)
    for i=1:KM
        Res=[Res,I(i,:)*Delta<=1,F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*
(Vmin-Vini)/periodo+Fo(:,i),F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta<=(60/T_tiempoReal)*
(Vmax-Vini)/periodo+Fo(:,i)];
    end
else
    for i=1:KM
        if i>ku
            Vinf(i,:)=Vmin;
            Vsup(i,:)=Vmax;
        else
            for j=1:Nt
                if (Vini(j)<=Vmax(j))
                    Vsup(i,j)=Vmax(j);
                else
                    Vsup(i,j)=(Vini(j)*(ku-i)+Vmax(j)*i)/ku;
                end
                if (Vini(j)>=Vmin(j))
                    Vinf(i,j)=Vmin(j);
                else
                    Vinf(i,j)=(Vini(j)*(ku-i)+Vmin(j)*i)/ku;
                end
            end
        end

        Res=[Res,I(i,:)*Delta<=1,F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*
(Vinf(i,:)-Vini)/periodo+Fo(:,i),F((i-1)*Nt+1:(i-1)*Nt+Nt,:)*Delta<=(60/T_tiempoReal)*
*(Vsup(i,:)-Vini)/periodo+Fo(:,i)];
    end
end

if resfinal==1
    Res=[Res,F((KM-1)*Nt+1:(KM-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(min(Vmax,Vini)-

```

16/10/19 17:25 D:\JONA\Traba...\optimiza bombeo futuro.m 4 of 5

```

Vini)/periodo+Fo(:,KM)]; %Restriccion que hace que el volumen final sea mayor o igual
que el inicial pero con una corrección por si empiezo fuera de los límites
elseif advmax>0
    Res=[Res,F((KM-1)*Nt+1:(KM-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*(min(Vmax,Vini)*
(1-excV)-Vini)/periodo+Fo(:,KM)];
end

if ResVmaxobj==true

    Res=[Res,F((indVmax-1)*Nt+1:(indVmax-1)*Nt+Nt,:)*Delta>=(60/T_tiempoReal)*
(Vmaxobj-Vini)/periodo+Fo(:,indVmax)];
end

if Resnumconm==true

    numcopeq=ceil(numconmmax*km*periodo/1440);
    if numcopeq<2 numcopeq=2; end
    Res=[Res,sum(abs(Yb*[Deltaini;Deltab(1:length(Deltab)-Nc)]))<=numcopeq];

end
if strcmp(solver,'mosek')
    sol=optimize(Res,J,sdpsettings('solver',solver,'mosek.MSK_DPAR_MIO_TOL_REL_GAP',
tolgap,'mosek.MSK_DPAR_MIO_MAX_TIME',maxtiempo))
else
    if strcmp(solver,'cbc')

        sol=optimize(Res,J,sdpsettings('solver',solver,'cbc.allowableFracGap',tolgap))
    else
        sol=optimize(Res,J,sdpsettings('solver','glpk','glpk.tmlim',60,'glpk.itlim',
50,'glpk.lpsolver',2,'glpk.branch',1,'glpk.round',0,'glpk.btrack',1))

    end
end

conmbo=sum(abs(Ybo*[Deltaini;value(Deltab(1:periodooptim/periodo*Nc-Nc)])))
Jo=T*Pg(:,1:periodooptim/periodo*Nc)*value(Deltab(1:periodooptim/periodo*Nc))

figure(2)
subplot(4,1,1)
hold off
plot(Volmedido);
axis([0,length(Volmedido),0 5]);
hold on
title('Volumes (in liters)');

subplot(4,1,2)
hold off
stairs([Qbombeo(1,:);Qbombeo(1,length(Qbombeo))]);
hold on
stairs([Qbombeo(2,:);Qbombeo(2,length(Qbombeo))]);
%Plot de las tarifas eléctricas
T2fl=[];
for ik=1:ndias

```

16/10/19 17:25 D:\JONA\Traba...\optimiza bombeo futuro.m 5 of 5

```

        T2f1=[T2f1,T2f];
    end
    T2f1=[T2f1,T2f(1:480)];
    stairs(10*[T2f1],'r:');
    %fin plot tarifas

    axis([0,length(Qbombeo),0 3]);
    title('Flow of filling');

    subplot(4,1,3)
    hold off
    plot(Qmedido(:,tinicial:tinicial+1440*ndias),'b');
    hold on
    plot((Fsalidaestimada),'r');
    axis([0,length(Qmedido),0 3]);
    title('Flow of emptying');

    subplot(4,1,4)
    hold off
    stairs([Valvula(1,:);Valvula(1,length(Valvula))'],'b:');
    hold on
    stairs([Valvula(2,:);Valvula(2,length(Valvula))'],'r--');
    axis([0,length(Valvula),0 1.5]);
    title('Valves');

```

1.4 FUNCIÓN: TIMER LAB

16/10/19 17:25 D:\JONA\Trabajos Johnata\S...\Timer Lab.m 1 of 2

```
function Timer_Lab()
global t;
global bandera;
global direcc_tiempoMatlab;

global direcc_nivel1;
global direcc_nivel2;
global Volmedido;
global Qmedido;
global Qbombeo;
global direcc_bol;
global direcc_bo3;
global Valvula;
global Deltaaplicada;
global Fp;
global Nc;
global Fsalidaestimada;
global perexpoQ;
global direcc_caudal2;
global direcc_caudal4;
global direcc_ev4;
global direcc_ev3;
global W;
global Ferror;

bandera = 1;
t = t+1;
m = modbus('tcpip', '192.168.1.3', 502);
write(m,'holdingregs' , direcc_tiempoMatlab, t);

    Nivel1_PLC = read(m, 'holdingregs' , direcc_nivel1,1);
    Nivel1_Litros = Nivel1_PLC*(4.867/32761);
    Nivel2_PLC = read(m, 'holdingregs' , direcc_nivel2,1);
    Nivel2_Litros = Nivel2_PLC *(4.44/32761);
Volmedido(:,t)= [Nivel1_Litros; Nivel2_Litros];%Volreal(:,t+1); %MEDICION DE LOS
NIVLES DE LOS DEPOSITOS
    QSalida1_PLC = read(m, 'holdingregs' , direcc_caudal2,1);
    QSalida2_PLC = read(m, 'holdingregs' , direcc_caudal4,1);
    QSalida1 = QSalida1_PLC*(8/16383);
    QSalida2 = QSalida2_PLC*(8/16383); %litros por minuto
Qmedido(:,t)= [QSalida1;QSalida2]; %medicion de los caudales de salida
Qbombeo(:,t)= Fp*Deltaaplicada((t-1)*Nc+1:t*Nc); %Caudal que bombeamos
    Qbombeo1 = Qbombeo(1,t);
    Qbombeo2 = Qbombeo(2,t);
    Qbombeo1_PLC = round(Qbombeo1*(16383/8));
    Qbombeo2_PLC = round(Qbombeo2*(16383/8));
    write(m,'holdingregs' , direcc_bol, Qbombeo1_PLC);
    write(m,'holdingregs' , direcc_bo3, Qbombeo2_PLC);
Valvula(:,t)=W*Deltaaplicada((t-1)*Nc+1:t*Nc); %Situación de las válvulas
for ij=1:2
    if Valvula(ij,t)==3
        Valvula(ij,t)=Valvula(ij,t-1); %Si el resultado de la optimización da que la
válvula este en 3, no importa su estado, se deja como estaba
    end
end
```

16/10/19 17:25 D:\JONA\Trabajos Johnata\S...\Timer Lab.m 2 of 2

```

    valvula4 = Valvula(1,t);
    valvula3 = Valvula(2,t);
    write(m,'holdingregs' , direcc_ev4, valvula4);
    write(m,'holdingregs' , direcc_ev3, valvula3);

Fsalidaestimada(:,1:1440)=Fsalidaestimada(:,2:1441); %Registro el nuevo valor estimado✓
del caudal de salida, corriendo en 1 el vector.
Error=Qmedido(:,t)-Fsalidaestimada(:,1); %Comparo el caudal medido de salida con el✓
caudal estimado y obtengo el error de mi estimación
Fsalidaestimada(:,1)=Qmedido(:,t); %Corrijo el valor actual con el que acabo de medir
Fsalidaestimada(:,1441)=Qmedido(:,t); %Corrijo tambien el de dentro de 24 horas puesto✓
que supongo que sigue un patron periodico
for i=2:perexpoQ*4
    Fsalidaestimada(:,i)=Fsalidaestimada(:,i)+Error*exp(-i/perexpoQ); %Corrijo✓
sumando una fracción del error cada vez en un tiempo perexpoQ=60 *4 de los siguientes✓
valores perexpoQ=60 *4 (4 horas)
end

end

```


2 LISTADO DEL CÓDIGO DE CODESYS

2.1 PROGRAMA: DEMANDA CAUDAL

```

0001 PROGRAM Demanda_Caudal
0002 VAR
0003
0004     periodosTranscurtidos: WORD := 1;
0005     a: WORD;
0006
0007     Timer1: TON;
0008     paso: BOOL := FALSE;
0009
0010     t: REAL := 300;
0011
0012     DR: TIME;
0013     BOMBILLA: BOOL := FALSE;
0014     a0: REAL;
0015     a1: REAL;
0016     W: REAL;
0017     a2: REAL;
0018     a3: REAL;
0019     b3: REAL;
0020     b1: REAL;
0021     b2: REAL;
0022     AUX: REAL;
0023
0024
0025     MAYOR: WORD;
0026     ACTUAL: WORD := 0;
0027     MENOR: WORD := 100;
0028     AUX2: REAL;
0029     CO2: WORD := 5;
0030
0031
0032     LimInf: REAL := 100000;
0033     LimSup: REAL := 0;
0034
0035     periodo_simulado AT %MW18 : WORD := 0; ("10 PERIODOS = 1 SEGUNDO")
0036     CAUDAL_MODEBUS AT %MW19 : WORD;
0037     CASO AT %MW20 : WORD;
0038
0039     ("Copia de control matlab")
0040     BO1_MAT %MW0: WORD;
0041     BO2_MAT %MW1: WORD;
0042     BO3_MAT %MW2: WORD;
0043     BO4_MAT %MW3: WORD;
0044     CAU1_MAT %MW4: WORD;
0045     CAU2_MAT %MW5: WORD;
0046     CAU3_MAT %MW6: WORD;
0047     CAU4_MAT %MW7: WORD;
0048     Nivel1_MAT %MW8: WORD;
0049     Nivel2_MAT %MW9: WORD;
0050     EV1_MAT %MW10: BOOL;
0051     EV2_MAT %MW11: BOOL;
0052     EV3_MAT %MW12: BOOL;
0053     EV4_MAT %MW13: BOOL;
0054     BO1_Lectura AT %MW14 : WORD;
0055     BO2_Lectura AT %MW15 : WORD;
0056     BO3_Lectura AT %MW16 : WORD;
0057     BO4_Lectura AT %MW17 : WORD;
0058
0059
0060     tiempo_matlab_anterior: WORD := 0;
0061     tiempo_matlab AT %MW21 : WORD;
0062
0063 END_VAR
0064
0065 VAR_EXTERNAL
0066
0067     REF_SALIDA: WORD;
0068     REF_SALIDA4: WORD;

```



```

0069
0070 BO1 AT %QW0 : WORD ;
0071 BO2 AT %QW1 : WORD ;
0072 BO3 AT %QW2: WORD ;
0073 BO4 AT %QW3: WORD ;
0074 CAU_1 AT %IW0: WORD;
0075 CAU_2 AT %IW1: WORD;
0076 CAU_3 AT %IW2: WORD;
0077 CAU_4 AT %IW3: WORD;
0078 NIVEL_1 AT %IW8: WORD;
0079 NIVEL_2 AT %IW5: WORD;
0080
0081 EV1 AT %QX4.0: BOOL;
0082 EV2 AT %QX4.1: BOOL;
0083 EV3 AT %QX4.3: BOOL;
0084 EV4 AT %QX4.2: BOOL;
0085 END_VAR

```

```

0001
0002 START := WORD_TO_BOOL(periodo_simulado);
0003
0004 IF START THEN
0005
0006     IF NOT (tiempo_matlab_anterior = tiempo_matlab) THEN
0007         t := tiempo_matlab;
0008         END_IF
0009         periodosTranscuridos := periodosTranscuridos + 1;
0010         t := t + (1/WORD_TO_REAL(periodo_simulado));
0011
0012     END_IF
0013
0014     a0 := 97.64;
0015     a1 := -38.18;
0016     a2 := -4.421;
0017     a3 := 14.34;
0018     b1 := 1.699;
0019     b2 := -17.57;
0020     b3 := 4.86;
0021     W := (2*3.14)/1440;
0022
0023     AUX := (a0 + a1*COS(W*t) + a2*COS(2*W*t)+a3*COS(3*W*t)+b1*SIN(W*t)+b2*SIN(2*W*t)+b3*SIN(3*W*t))/30;
0024     AUX := AUX * 820;
0025
0026     REF_SALIDA := REAL_TO_WORD(AUX*0.9);
0027     REF_SALIDA4 := REAL_TO_WORD(AUX);
0028
0029
0030
0031     Refbo1 := BO1_Mt
0032     Refbo2 := BO2_Mt
0033     Refbo3 := BO3_Mt
0034     Refbo4 := BO4_Mt
0035
0036     BO1_Lectura:= BO1;
0037     BO2_Lectura:= BO2;
0038     BO3_Lectura:= BO3;
0039     BO4_Lectura:= BO4;
0040
0041     CAU1_M := CAU_1;
0042     CAU2_M := CAU_2;
0043     CAU3_M := CAU_3;
0044     CAU4_M := CAU_4;
0045
0046     Nivel1_M := NIVEL_1;
0047     Nivel2_M := NIVEL_2;
0048
0049     EV1 := EV1_Mt;
0050     EV2 := EV2_Mt;
0051     EV3 := EV3_Mt;

```

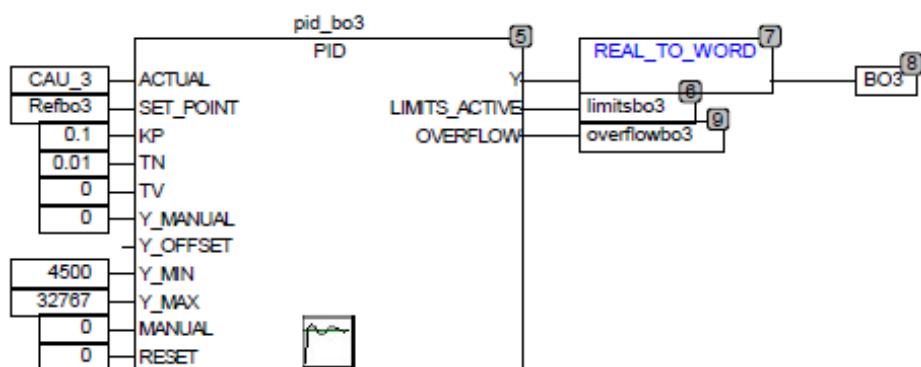
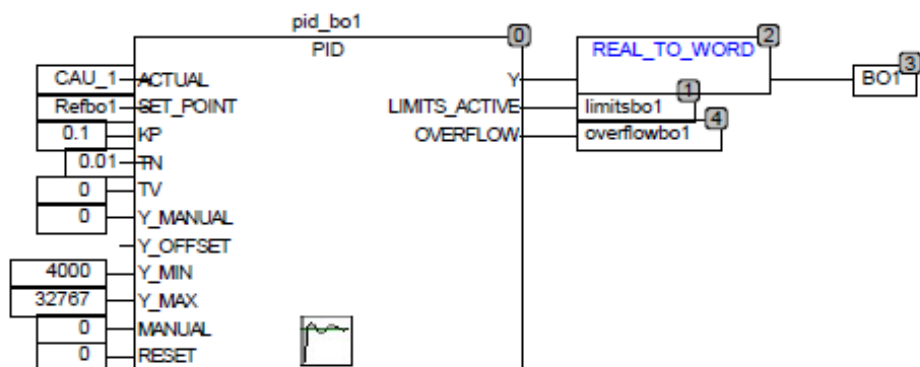
0052	EV4 := EV4_M;
0053	
0054	tiempo_matlab_anterior := tiempo_matlab;
0055	

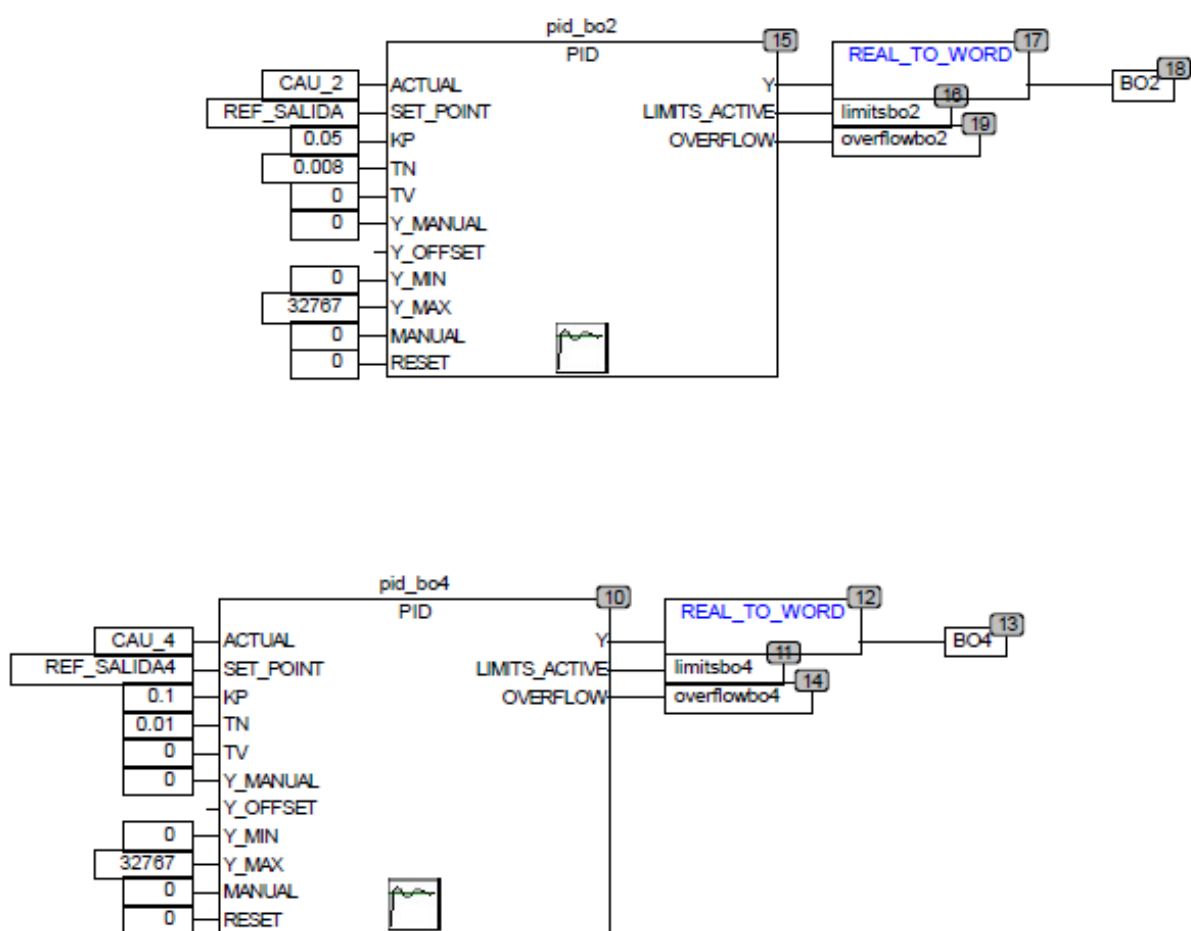
2.2 PROGRAMA: CONTROL BOMBAS

```

0001 PROGRAM Control_Bombas
0002 VAR
0003
0004   pid_bo1: PID;
0005   pid_bo3: PID;
0006   pid_bo4: PID;
0007   pid_bo2: PID;
0008   pid_bo1_t2: PID;
0009   pid_bo3_t1: PID;
0010 END_VAR
0011
0012 VAR_EXTERNAL
0013
0014   REF_SALIDA: WORD;
0015   REF_SALIDA4: WORD;
0016
0017 END_VAR

```



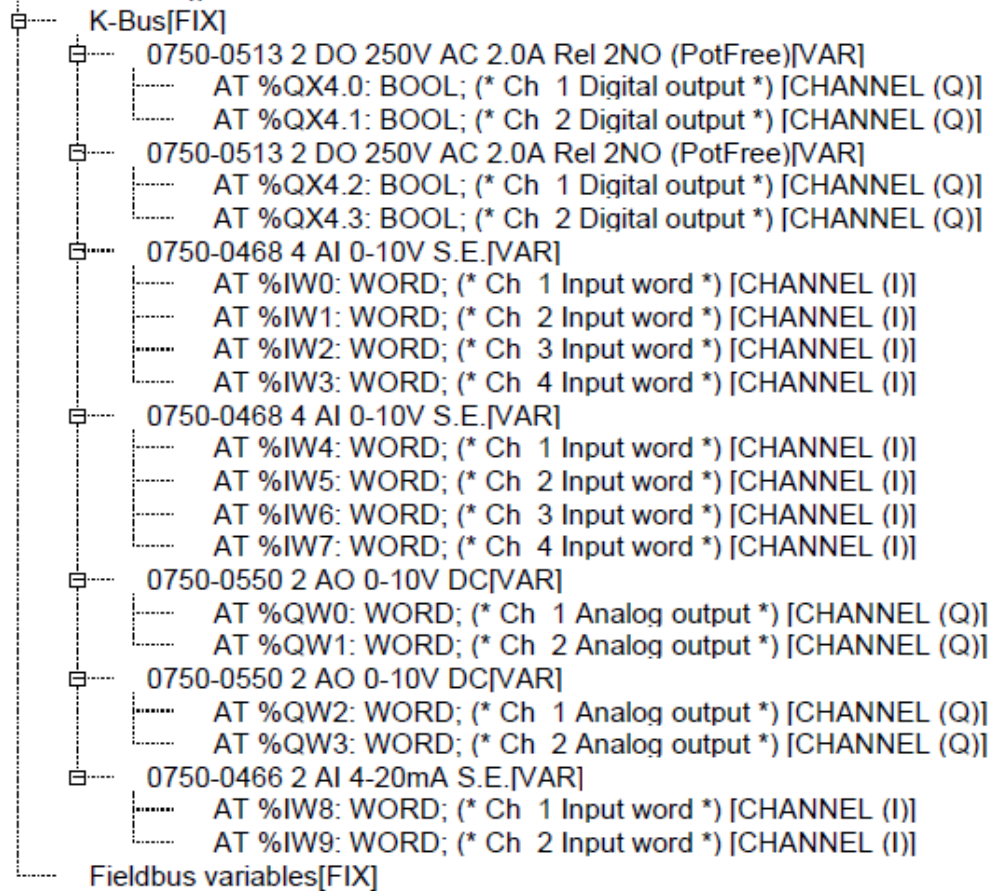


2.3 VARIABLES GLOBALES

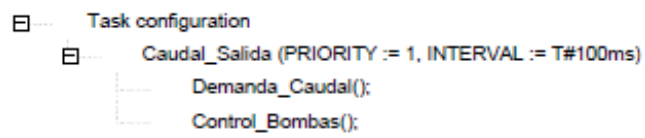
0001	VAR_GLOBAL
0002	
0003	
0004	BO_AUX: WORD;
0005	BO_AUX2: WORD;
0006	EV1 AT %QX4.0: BOOL;
0007	EV2 AT %QX4.1: BOOL;
0008	EV3 AT %QX4.3: BOOL;
0009	EV4 AT %QX4.2: BOOL;
0010	
0011	COPIA1: WORD;
0012	COPIA2: WORD;
0013	COPIA3: WORD;
0014	COPIA4: WORD;
0015	
0016	BO1 AT %QW0 : WORD ;
0017	BO2 AT %QW1 : WORD ;
0018	BO3 AT %QW2: WORD ;
0019	BO4 AT %QW3: WORD ;
0020	CAU_1 AT %IW0: WORD;
0021	CAU_2 AT %IW1: WORD;
0022	CAU_3 AT %IW2: WORD;
0023	CAU_4 AT %IW3: WORD;
0024	NIVEL_1 AT %IW8: WORD;
0025	NIVEL_2 AT %IW5: WORD;
0026	
0027	Refbo1: WORD;
0028	Refbo2: WORD;
0029	Refbo3: WORD;
0030	Refbo4: WORD;
0031	Refbo1_T2: WORD;
0032	Refbo3_T1: WORD;
0033	
0034	limitsbo1: BOOL;
0035	overflowbo1: BOOL;
0036	limitsbo2: BOOL;
0037	overflowbo2: BOOL;
0038	limitsbo3: BOOL;
0039	overflowbo3: BOOL;
0040	limitsbo4: BOOL;
0041	overflowbo4: BOOL;
0042	limitsbo1_T2: BOOL;
0043	overflowbo1_T2: BOOL;
0044	limitsbo3_T1: BOOL;
0045	overflowbo3_T1: BOOL;
0046	
0047	REF_SALIDA: WORD;
0048	REF_SALIDA4: WORD;
0049	
0050	START: BOOL := 0;
0051	STOP: BOOL := 0;
0052	
0053	
0054	END_VAR

2.4 CONFIGURACIÓN E/S PLC

Hardware configuration



2.5 CONFIGURACIÓN DE TAREAS



3 FICHAS TÉCNICAS

3.1 BOMBA SUMERGIBLE

COMET-Pumpen

Systemtechnik GmbH & Co. KG

Industriestraße 5

37308 Pfaffschwende

Telefon +49 (0) 3 60 82/4 38-0

Telefax +49 (0) 3 60 82/4 36 34

E-Mail: kontakt@comet-pumpen.de

Internet: http://www.comet-pumpen.de



Tauchpumpen / submersible pumps

ELEGANT 24V

Artikel: 1305.79.00

Tauchpumpe ELEGANT 24V

submersible pump ELEGANT 24V

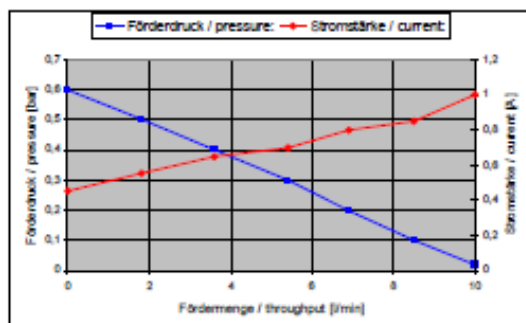


Spannung / voltage:	24 Volt = / 24 volt DC
Stromstärke / current:	max. 1 A
Fördermenge / throughput:	max. 10 l/min
Förderhöhe / delivery height:	max. 5 m
Förderdruck / pressure:	max. 0,5 bar / 7,2 psi
Verbrauch / power:	15 - 25 Watt
Ø / Höhe / diam. / height:	38 / 104 mm
Kabellänge / wire length:	1 m
Anschluss / connection:	(+) braun / brown (-) blau / blue

- trockenlaufsicher bis 2 Stunden
- dauerlaufgeeignet
- Laufzeit bei Betrieb mit 12V ca. 1000 h, sonst 500 h
- zur Förderung von sauberem Wasser bis max. 60°C, inklusive Trinkwasser
- für Schläuche mit 10 mm Innendurchmesser
- erweiterbar mit Rückschlagventil, Entlüftungsstutzen und Filter
- andere Kabellängen auf Anfrage lieferbar

- run dry up to 2 h without damage
- suitable for continuous operation
- running time 1000 h when operating with 12V, otherwise 500 h
- for supply of clean water up to 60°C, including drinking water
- for hoses with 10 mm inner diameter
- expandable with non-return-valve, vent connection and filter
- other cable length available on request

Förderdiagramm / capacity diagram:



Varianten / variants:

Tauchpumpe ELEGANT 24V mit Entlüftungsstutzen / submersible pump ELEGANT 24V with vent connection
Art.-Nr.: 1315.79.00

Tauchpumpe ELEGANT 24V mit entlüftetem Rückschlagventil und Filter / submersible pump ELEGANT 24V with check valve vent and filter

Art.-Nr.: 1325.79.00

Zubehör / accessories:



Entlüftungsstutzen / vent connection
C1.0905.01



Rückschlagventil mit Entlüftung / non-return-valve with venting
3004.53.00



Filter
3092.13.00

3.2 AMPLIFICADOR



DC MOTOR CONTROLLER MODULE

DCM

X10335

INTRODUCTION

This general purpose, modulated, pulse-width, low voltage dc controller, can be operated in any of the following modes:

Motor Control: High Frequency (RT/RT1 no link) speed control set by a 5k Ω potentiometer.

Lighting/Heating Control: Low frequency (RT/RT1 linked) output level set by a 5k Ω potentiometer as above.

Temperature Control: Thermistor connected across RT/RT1, with a temperature range of 5-130°C. Temperature set by a 5k Ω potentiometer.

APPLICATIONS

Include speed control of low voltage, high frequency, dc motors, low voltage lighting and medium frequency heaters.

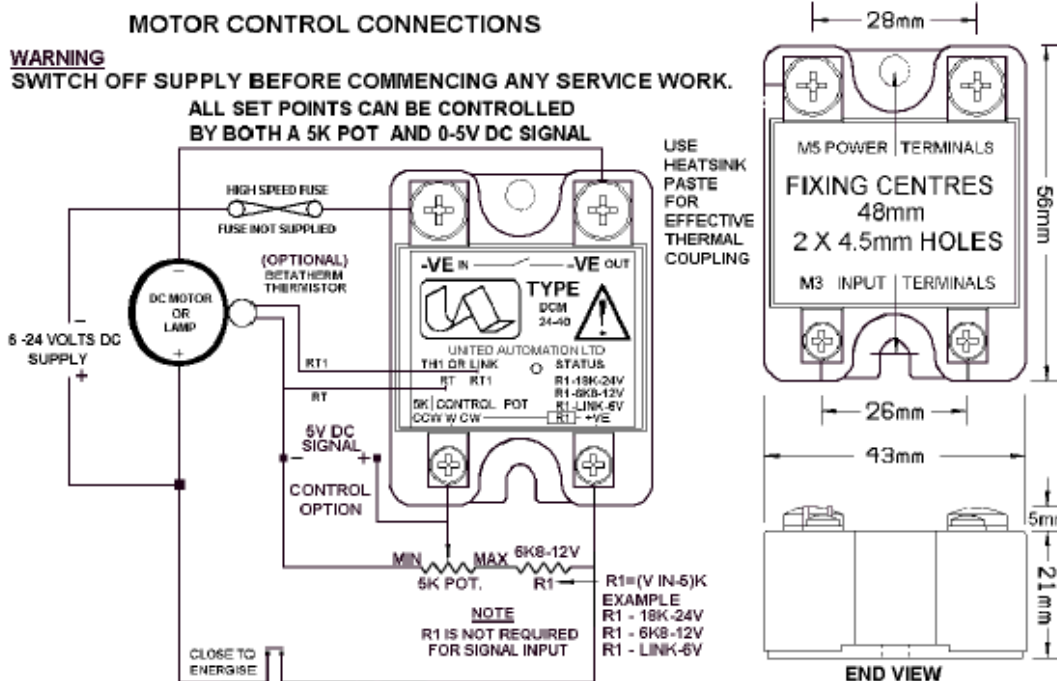
FEATURES

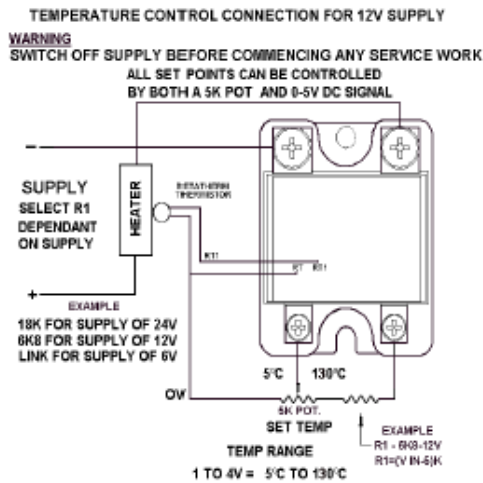
- Manual or signal control.
- Temperature control with optional sensor.
- 180 or 350Hz selectable frequency ranges.
- Short-circuit protection.
- 6 to 24V dc supply voltage range.

RoHS Compliant



INSTALLATION



INSTALLATION**PROTECTION NOTE:**

For controller protection a 'TRANSIL' component device is recommended to be fitted (hard wired) across the following supply terminals – 'M3 +ve' and 'M5 –ve IN'.

SPECIFICATIONS

Maximum dc system line voltage

Unit limiting dc current

Control input voltage range

Control input current @ 5V typical

High frequency mode (no link across RT and RT1)

Medium frequency mode (link RT and RT1)

Optional for temperature control (terminals RT & RT1): Thermistor type- Betatherm - 10K3A1

Unit operating temperature range

Unit storage temperature range

24V dc

40A dc

0-5V dc

1mA dc

350Hz

180Hz

5 - 130°C

0 to 85°C

0 to 85°C

FUSING

It is recommended that semiconductor, fast-acting type fuses or circuit breakers (semiconductor - MCB) be used for unit/device protection. On initial operation some loads may need an increased factor of safety for unit/device protection (see SRA datasheet for further information).

CE MARKING

This product family carries a CE marking. For information see recommendation section and contact our sales des. (see Declaration of Conformity).

RECOMMENDATION

Other documents are available on request, which may be appropriate for your applications.

CODE	IDENTITY	DESCRIPTION
X10229	RFI	Filter recommendation: Addressing the EMC directive.
X10213	ITA	Interaction: Uses for phase angle and for burst fire control.
X10255	SRA	Safety requirements: addressing the Low Voltage Directive (LVD) including: thermal data/cooling, live parts warning, earth requirements and fusing recommendations.
P01.1	COS	UAL conditions of sale.

NOTE: It is recommended that installation and maintenance of this equipment should be done with reference to the current edition of the I.E.T. (formerly I.E.E.) regulations (BS7671) by suitably qualified/trained personnel. The regulations contain important requirements regarding installation and safety of electrical equipment. Specific installers should refer to local and national regulations.

ORDER CODE:

Optional extras include:-

Further extras include:-

State part number: DCM -24-40

Betatherm 10K3A1 bead sensor only –

Betatherm 10K3A1 bead (type-X) sensor with 1m PTFE leads:

Betatherm 10K3A1 enclosed (type-E) sensor with 1m PTFE leads

Heatsink assemblies for 40A capability; Heat sink paste; 5K potentiometer.

Stock code D80005

Stock code A26048

Stock code A26038

**UNITED AUTOMATION LIMITED**

Southport Business Park
Wight Moss Way
Southport, PR8 4HQ
ENGLAND

Tel: 0044 (0) 1704 – 516500
Fax: 0044 (0) 1704 – 516501
enquiries@united-automation.com
www.united-automation.com

Page No. 2 of 2

Issue 6

Date 31/10/14

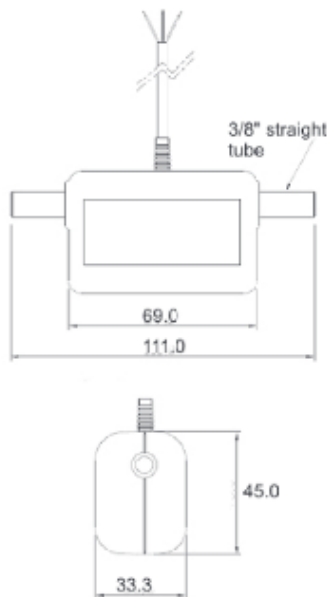


3.3 CAUDALÍMETRO

Ultrasonic Flow Meter UF08B



- Non-invasive sensor technology
- Fully electronic, no moving parts
- Automatic viscosity and temperature compensation
- Pulse or analogue output selectable
- Low pressure drop
- LED indication of no liquid & of flow rate



Cynergy3 Components Ltd.
7 Cobham Road
Ferndown Industrial Estate
Wimborne, Dorset BH21 7PE
Telephone +44 (0) 1202 897969

Email: sales@cynergy3.com

ISO9001 CERTIFIED

This innovative design provides a high accuracy, non-invasive, flow measurement device at a fraction of the cost of other current non-invasive systems. The unique measurement technique automatically compensates for viscosity and temperature variations. The measurement of flow is by ultrasonic transit time in-line cell.

The flow path is designed to minimise pressure drop and, having no moving parts within, will not clog or jam. The sensor also allows contaminants to pass through without affecting its performance.

The sensor is supplied with pulse and analogue outputs, which are selectable by connection wires..

Technical Specifications

Max. flow L/min (Q4)	8
Transitional flow (Q2)	0.4
Min. flow L/min (Q1)	0.1
Output selectable	Pulse Analogue

Performance

Accuracy for flow rates between Q2 and Q4	3% of reading
Accuracy for flow rates between Q1 and Q2	5% of reading
Resolution better than	0.001L/min
Reverse flow	0-8L/min
Response time	Better than 0.1s

Interface

Connection 8 wires (RED supply +ve, BLACK ground, Blue NPN output, Yellow Data in, White PNP output, Orange Voltage output, Brown Current output, Green Ground connected internally to black)

Supply	8 - 24VDC (input current <15mA @ 24VDC)
Output selectable	1000 pulses/L 0-5Vdc 4-20mA

Operation

Principle	Ultrasonic transit time in-line flow cell
Temp. range (fluid)	-10°C to 85°C
Continuous fluid sound	Maintains performance regardless of fluid type, temperature or viscosity for speed measurement fluids with sound speeds 1250 - 1750 m/s

Physical characteristics

Flow tube material	Glass filled plastic, Grivory HTV-4FWA Black 9225 (FDA and EU approved for foodstuffs)
Flow tube internal diameter	7mm
Connection thread	3/8" straight tube
Internal bore of connection	7mm
Suitable Pushfit adaptor (to fit 3/8" OD Tube)	John Guest Speedfit P10412S
Maximum pressure	10 bar
Case material	ABS black, Polylac PA-757
Case integrity	Ultrasonically welded, not liquid proof
Connection	8 core, PVC sheathed, 100cm long standard
Environmental protection	Internal electronics conformally coated.

Ordering Code

Series	UF 08 B 100	Cable length (cms)
Flow range (8= 0 to 8L/min)		Output (B = selectable)

www.cynergy3.com

© 2011 Cynergy3 Components, All Rights Reserved. Specifications are subject to change without prior notice. Cynergy3 Components and the Cynergy3 Components logo are trademarks of Cynergy3 Components Limited.

3.4 ELECTROVÁLVULA

Inline Sprinkler Control Valve

Heavy-Duty Self Cleaning Design—For years of reliable service

Fail-Safe Reverse-Flow Design—Prohibits the free flow of water (due to a ruptured diaphragm) reducing potential water waste and flood damage

Manual On/Off Bypass Lever—Ideal for testing and flushing sprinkler lines

Universal—Works with residential standard 24-volt sprinkler systems

57100 3/4" FNPT without Flow Control

57101 1" FNPT without Flow Control

57105 1" Slip without Flow Control

Inline Sprinkler Control Valve— With Flow Control Additional Features

Flow Control Adjustment—Reduces sprinkler fogging and over-spray

57102 3/4" FNPT with Flow Control

57103 1" FNPT with Flow Control

Operation Range:

Pressure:
3/4" & 1" 15-150 psi

Flow Rate: 3/4" 5 to 35 gpm
1" 5 to 40 gpm

Electrical Specifications:

24 VAC 50/60 cycle solenoid power requirement: 0.30 A (7.2 VA) inrush current; 0.19 A (4.6 VA) holding current

Mechanical Specifications:

Dimensions:

Height: 5-1/2"

Width: 4"

Length: 4-1/2"

Environmental Specification:

Temperature:

Up to 158° F (70° C)

* Orbit recommends flow rates at or below 5 feet per second on all valve models.

Standard		
Valve Pressure Loss (psi)		
Flow gpm	3/4" psi	1" psi
5	2.5	3
10	3	3.5
15	4	4.5
20	5.5	5.5
25	9.5	7
30	13	10
35	16.5	12.5
40	-	15

3.5 SENSOR DE NIVEL ENDRESS HAUSER

Datos técnicos

Entradas	
Viscosidad máxima	2000 cSt
Campo de medida	0 a 2000 pF
Longitud de la sonda	150 a 2500 mm
Span tolerado	$\Delta C = 10 \text{ pF a } 2000 \text{ pF}$
Frecuencia de medición	250 kHz
Señal de entrada	Sondas sumergidas => capacidades altas Sondas al aire => capacidades bajas
Salida (electrónica FEI20/4 a 20mA)	
Señal de salida	3,8 a 20,5 mA
Corriente de activación	máx. 20 mA (<500 ms)
Señal de alarma	>21 mA
Fuente de alimentación	
Tensión para la conex.	U = 10 a 30 V CC, protección contra inversión de polaridad
Consumo	P <0,7 W
Consumo de corriente	I <22 mA
Entradas para cable	M20 x 1,5 (conexiones a rosca)
Características de ejecución (con la electrónica inserta instalada)	
Condiciones de funcionamiento de referencia	Temp. ambiente 23 °C, presión atmosférica, instalación vertical de la sonda desde arriba
Error de medición máx.	±1 % del valor de fondo de escala
Repetibilidad	0,25 % del valor de fondo de escala
Tiempo de reacción	<2 s
Influencia de la temp. ambiente	<0,01 %/K (-40 a +70 °C) longitud de la sonda 1 m
Tiempo de respuesta	1 s
Calibración de fábrica	Una vez instalado, un recalibrado sólo es necesario si: <ul style="list-style-type: none"> ▪ hay que ajustar los valores de 0 % y 100 % a las especificaciones particulares del cliente ▪ en caso de haber reducido la longitud de las varillas de la sonda

Condiciones de aplicación

Temp. ambiente	-40 a +70 °C
Temp. almacenamiento	-40 a +80 °C
Clase climática	Apto para el clima típico de los trópicos según DIN IEC 68 Parte 2-38
Grado de protección	IP 66
Resistencia a golpes	DIN EN 60068-2-27/IEC 68-2-27: 30g
Resist. a vibraciones	DIN EN 60068-2-64/IEC 68-2-64: 20 a 2000 Hz, 1 (m/s ²) ² /Hz (con longitud mín. para las varillas de 150 mm)
CEM	Emisión de interferencias según EN 61326, clase B para equipos eléctricos; Inmunidad ante interferencias según EN 61326, Anexo A (equipos industriales)
Conductividad del medio	≥30 µS/cm
Presión de proceso	-1 a +10 bar
Temp. de proceso	-40 a +100 °C
Resistencia a cargas laterales	2 Nm

Materiales en contacto con el medio

Varillas de la sonda	Varilla: 1.4404/316L; Opcional: fibra de carbono CFC; Junta: EPDM; Aislante: PP; Separador: PP
Conexiones a proceso	G1½ A (PPS, DIN ISO 228/1)
Juntas	Junta anular para una conexión a proceso G1½ A; Elastómero exento de fibra de amianto (a prueba de aceites, disolventes, vapor, ácidos y bases débiles)

Indicador

LED verde	modo operativo (parpadeo lento), modo de calibración (parpadeo rápido)
LED rojo	para validación desde teclado (parpadeo corto), alarma o aviso (destello)
Indic. de valor medido en %	opcional

Certificaciones

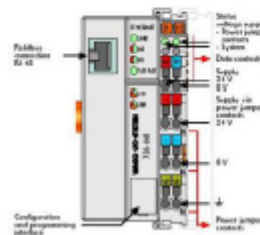
Certificación WHG	protección contra rebose según § 19 WHG (Alemania)
-------------------	--

3.6 PLC

Data sheet from 16.06.2010



Item no.:	750-841
Product description:	ETHERNET TCP/IP PROGRAMMABLE FIELDBUS CONTROLLER 10/100 Mbit/s DIGITAL AND ANALOG SIGNALS



Packing unit 1 piece

Product group	15 (I/O System)
No. of controllers connected to the Master	limited by ETHERNET- Specification
Transmission medium	Twisted Pair S-UTP 100 Ohm Cat 5
Max. length of fieldbus segment	100 m between Hub and 750-841; max. length of network limited by Ethernet specifications
Baud rate	10/100/Mbit/s
Buscoupler connection	RJ45
Protocols	MODBUS / TCP (UDP), ETHERNET / IP, HTTP, BootP, DHCP, DNS, SNTP, FTP, SNMP, SMTP
Programming	WAGO-I/O-PRO CAA
IEC 61131-3	AWL, KOP, PUP, ST, AS
Max. no. of I/O modules	64
with bus extension	250
Input variables	512 Byte
Output variables	512 Byte
Configuration	through the computer
Program memory	512 kByte
Data memory	256 kByte
Non-volatile memory (retain)	24 Kbyte (16 k retain, 8 k flag)
Voltage supply	DC 24 V (-15% / + 30%)
Input current (max)	500 mA
Efficiency of the power supply	87 %
Internal current consumption	300 mA
Total current for the I/O modules	1700 mA

Data sheet from 16.06.2010



Isolation	500 V system / supply
Voltage via power jumper contacts (max)	DC 24 V (-15% / + 30%)
Current via power jumper contacts (max)	10
Operating temperature	0 °C ... + 55 °C
Storage temperature	-25 °C ... +85 °C
Relative air humidity	95 %
Vibration resistance	acc. IEC 60068-2-6
Shock resistance	acc. IEC 60068-2-27
Degree of protection	IP 20
EMC immunity to interference	gem. EN 61000-6-2 (1999)
EMC emission of interference	acc. EN 50081-2 (94)
EMC marine applications - noise proof	gem. Germanischer Lloyd (2003)
EMC marine applications - noise transmission	gem. Germanischer Lloyd (2003)
Cross section [mm²]	0.08 - 2.5 mm²
Cross section [AWG]	28 - 14 AWG
Weight	184 g
Color	light gray
Height	65 mm
Height	2.56 in
Width	51 mm
Width	2.008 in
Depth	100 mm

Item no.	Admission office	Approval no.	Voltage [V]	Current [A]	Cross section [mm²]
750-841	ABS	03-HG374860/2-PDA			
750-841	BSH	Bescheinigung Nr. 467			
750-841	BV	13453/B0 BV			
750-841	DNV	A-10191			
750-841	GL	26 898 - 05 HH			
750-841	IECEx	IECEx PTB 07.0064X			
750-841	KR	HMB06880-EL006			
750-841	LR	02/20026 (E2)			
750-841	NK	TA06190M			
750-841	PRS	TE/1720/880590/08			
750-841	Rina	ELE153207CS001			
750-841	UL	E175199 Sec.1			
750-841	UL	E198726 Sec.1			
750-841	UL DEMKO A/S	DEMKO 08 ATEX 142851 X			

Sorting criteria: admission office - approval number

© WAGO Kontakttechnik GmbH & Co. KG

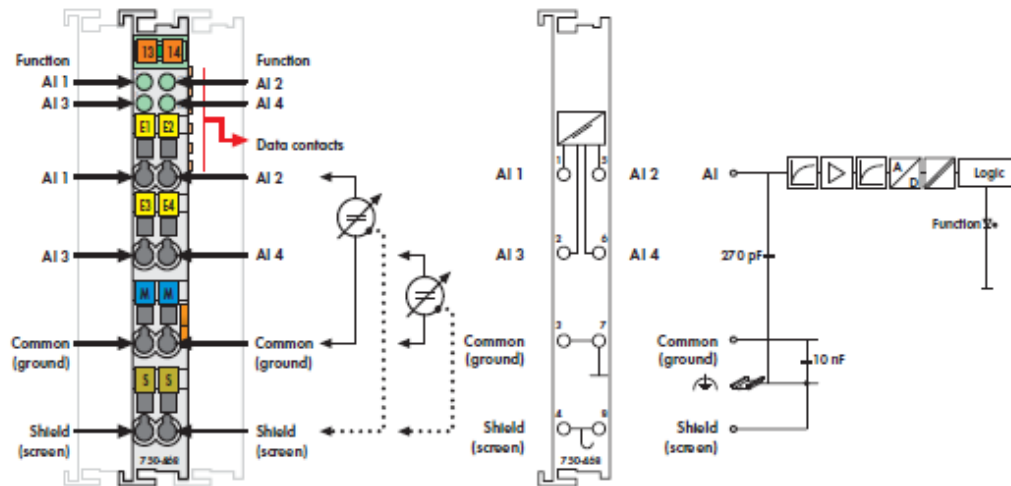
Specifications are subject to changes and errors may be expected

3.7 MODULOS WAGO

750-468







4-Channel Analog Input Module 0-10 V

Single-ended (S.E.)



Delivered without miniature WSB markers

The analog input module receives signals with the standardized values of 0...10V.
The input signal is electrically isolated and will be transmitted with a resolution of 12 bits.
The internal system voltage supply is used for the power supply of the module.
The input channels of a module have one common ground potential.
The shield (screen) is directly connected to the DIN rail.

Description	Item No.	Pack Unit
4AI 0-10V DC S.E.	750-468	1
4AI 0-10V DC S.E. SS ¹⁾	750-468/000-200	1
4AI 0-10V DC S.E./T	750-468/025-000	1
Extended temperature range: -20 °C ... +60 °C		
¹⁾ Data format for SS control with FB 251		
Accessories	Item No.	Pack Unit
Miniature WSB Quick marking system		
 plain	248-501	5
 with marking	see Section 11	
Approvals		
Conformity marking	CE	
Korea Certification		
Marine applications (versions upon request)	ABS, BV, DNV, GL, KR, LR, PRS, RINA	
 UL 508		
 ANSI/ISA 12.12.01	Class I, Div. 2, Grp. A,B,C,D, T4	
 TÜV G7 ATEX 554086 X	I M2 Ex d I Mb, II 3 G Ex nA IIC T4 Gc, II 3 D Ex tc IIIC T135°C Dc Ex d I Mb, Ex nA IIC T4 Gc, Ex tc IIIC T135°C Dc	
IECEx TUN 09.0001.X		

Technical Data	
Number of inputs	4
Power supply	via system voltage DC/DC
Current consumption typ. (internal)	60 mA
Input voltage (max.)	35 V
Signal voltage	0 ... 10 V
Internal resistance	133 kΩ
Resolution	12 bits
Conversion time (typ.)	4 ms
Measuring error (25 °C)	± 0.2 % of the full scale value
Temperature coefficient	< ± 0.01 % / K of the full scale value
Isolation	5000 V system/supply
Btt width	4 x 16 bits data
	4 x 8 bits control/status (optional)
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm² ... 2.5 mm² / AWG 28 ... 14
Strip lengths	8 ... 9 mm / 0.33 in
Width	12 mm
Weight	52.5 g
EMC immunity of interference	acc. to EN 61000-6-2, marine applications
EMC emission of Interference	acc. to EN 61000-6-4, marine applications

750-550, 750-556 / 753-550, 753-556

2-Channel Analog Output Module 0-10 V/±10V

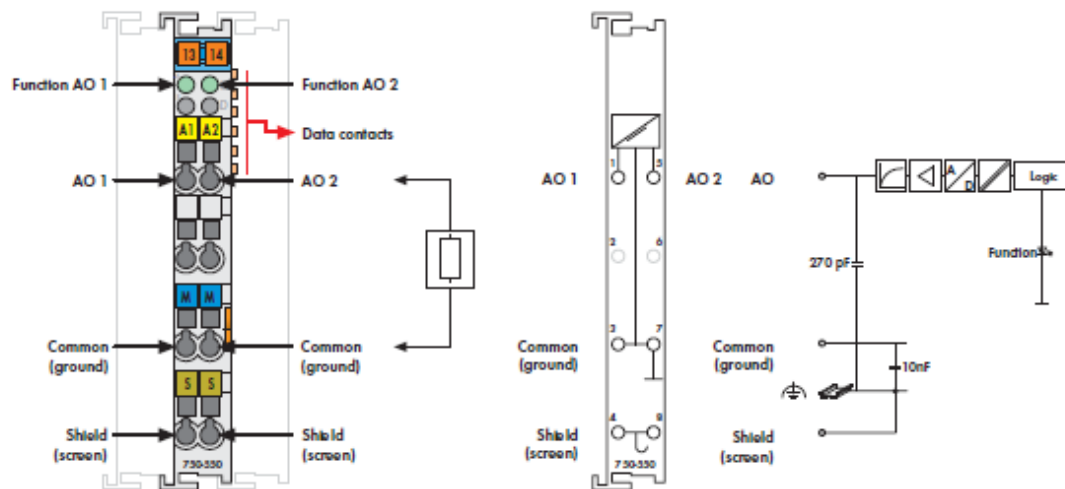









Fig. 750 Series
Delivered without miniature WSB markers

The analog output module creates a standardized signal of 0-10V or ±10V. The output channels have one common ground potential.

The output signal is electrically isolated and will be transmitted with a resolution of 12 bits.

Outputs are short circuit protected.

The internal system supply is used for the power supply of the module.

Description	Item No.	Pack. Unit
2AO 0-10V DC	750-550	1
2AO ±10V DC	750-556	1
2AO 0-10V DC/SS ¹⁾	750-550/000-200	1
2AO ±10V DC/SS ¹⁾	750-556/000-200	1
2AO 0-10V DC (without connector)	753-550	1
2AO ±10V DC (without connector)	753-556	1
¹⁾ Data format for SS control with FB 251		
Accessories	Item No.	Pack. Unit
 753 Series Connectors	753-110	25
 Coding elements	753-150	100
 Miniature WSE Quick marking system plain	248-501	5
with marking	see Section 11	
Approvals		
Conformity marking	CE	
Korea Certification		
Marine applications (versions upon request)	ABS, BV, DNV, GL, KR, LR, NK, PRS, RINA	
 UL 508		
 ANSI/ISA 12.12.01	Class I, Div. 2, Grp. ABCD, T4	
 TÜV 07 ATEX 554086 X	I M2 Ex d I Mb, II 3 G Ex nA IIC T4 Gc, II 3 D Ex tc IIC T135°C Dc	
IECEX TUN 09.0001 X	Ex d I Mb, Ex nA IIC T4 Gc, Ex tc IIC T135°C Dc	

Technical Data	
No. of outputs	2
Current consumption (Internal)	65 mA
Power supply	via system voltage DC/DC
Signal voltage	0 - 10V (750-550 / 753-550) ± 10V (750-556 / 753-556)
Load impedance	> 5 kΩ
Linearity	±10 mV
Resolution	12 bits
Conversion time	approx. 2 ms
Recovery time (typ.)	300 μs
Measuring error (25 °C)	< ± 0.1 % of the full scale value
Temperature coefficient	< ± 0.01 % /K of the full scale value
Isolation	500 V system/supply
Bit width	2 x 16 bits data 2 x 8 bits control/status (option)
Wire connection	CAGE CLAMP®
Cross sections	0.08 mm² ... 2.5 mm² / AWG 28 ... 14
Strip lengths, 750/753 Series	8 ... 9 mm / 0.33 in 9 ... 10 mm / 0.37 in
Width	12 mm
Weight	48.8 g
EMC immunity of interference	acc. to EN 61000-6-2, marine applications
EMC emission of interference	acc. to EN 61000-6-4, marine applications

750-513 / 753-513

2-Channel Relay Output Module 230 V AC, 30 V DC

Isolated outputs; 2 make contacts

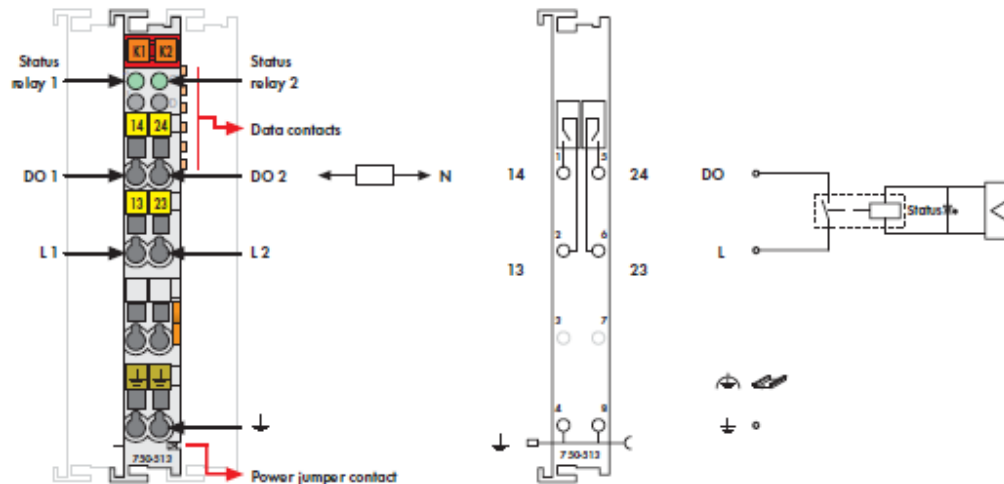


Fig. 750 Series
Delivered without miniature WSB markers

Control signals are transmitted from the automation device to connected actuators via the digital output module.

The internal system voltage is used to trigger the relay.

The NO contacts are electrically isolated.

The switched status of the relay is shown by an LED.

Description	Item No.	Pack. Unit
2DO 230V AC 2.0A/ Relay 2NO/ Potential Free	750-513	1
2DO 230V AC 2.0A/ Relay 2NO/ Potential Free / without power jumper contacts	750-513/ 000-001	1
2DO 230V AC 2.0A/ Relay 2NO/ Potential Free (without connector)	753-513	1
Accessories	Item No.	Pack. Unit
753 Series Connectors	753-110	25
Coding elements	753-150	100
Miniature WSB Quick marking system plain	248-501	5
with marking	see Section 1.1	
Approvals		
Conformity marking	CE	
Korea Certification	KS	
Marine applications (versions upon request)	ABS, BV, DNV, GL, KR, LR, NKK, PRS, RINA	
UL 508		
ANSI/ISA 12.12.01	Class I, Div. 2, Grp. ABCD, T4	
TÜV 07 ATEX 554086 X	I M2 Ex d I Mb	
	II 3 G Ex nA nC IIC T4 Gc	
	II 3 D Ex tc IIC T135°C Dc	
IECEx TUN 09.0001 X	Ex d I Mb	
	Ex nA nC IIC T4 Gc	
	Ex tc IIC T135°C Dc	
Technical Data		
No. of outputs	2 make contacts	
Max. current consumption (internal)	100 mA	
Max. switching voltage	250 V AC / 30 V DC	
Switching power	500 VA / 60 W (resistive load)	
	cos ϕ max = 0.4; I/R max = 7 ms	
Min. switching current	10 mA / 5 V DC	
Max. switching current	2 A AC/DC	
Max. switching frequency	30/min (at nominal load)	
Pull-in time (max.)	10 ms	
Bounce time (typ.)	1.2 ms	
Drop-out time (max.)	10 ms	
Contact material	Silver alloy	
Mechanical life (min.)	2 x 10 ⁷ switching operations	
Electrical life (min.)	3 x 10 ⁵ switching operations	
	(2 A / 250 V AC) or (2 A / 30 V DC)	
Isolation	1.5 kV off (field/system)*;	
	* 2.5 kV rated surge voltage	
Internal bit width	2 bits	
Wire connection	CAGE CLAMP®	
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14	
Strip lengths, 750/753 Series	8 ... 9 mm / 0.33 in	
	9 ... 10 mm / 0.37 in	
Width	12 mm	
Weight	53.5 g	
EMC immunity of interference	acc. to EN 61000-6-2, marine applications	
EMC emission of interference	acc. to EN 61000-6-4, marine applications	

WAGO Kontakttechnik GmbH & Co. KG
Subject to design changes

26.01.2016

Postfach 2880 · D-32385 Minden
Hansastr. 27 · D-32423 Minden

Tel.: +49(0)571/887-0 E-Mail: info@wago.com
Fax: +49(0)571/887-169 www.wago.com

750-465, 750-466 / 753-465, 753-466

2-Channel Analog Input Module 0/4-20 mA

Single-ended (S.E.)

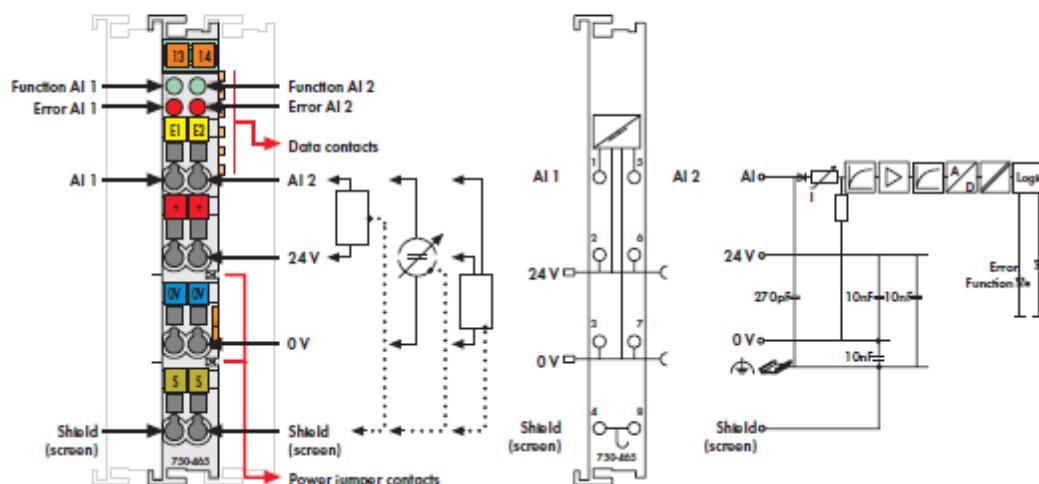


Fig. 750 Series/Technical data see page 24/Delivered without miniature WSB markers
750/753 Series marking see pages 10 ... 11 / 12 ... 13

The analog input module powers the transducers, receives transmitted analog signals, and with electrical isolation, transmits the signals to the fieldbus.

The 24V supply for the field is derived from the module's power jumper contacts.

The shield (screen) is directly connected to the DIN rail.

This input module can supply the voltage for 2-wire transducers.

Description	Item No.	Pack. Unit
2A1 0.20mA S.E.	750-465	10 ¹¹
2A1 0.20mA S.E. S5 ¹⁾	750-465/000-200	1
2A1 0.20mA S.E./T	750-465/025-000	1
(Operating temperature -20 °C ... +60 °C)		
2A1 4.20mA S.E.	750-466	10 ¹¹
2A1 4.20mA S.E. S5 ¹⁾	750-466/000-200	1
2A1 4.20mA S.E./T	750-466/025-000	1
(Operating temperature -20 °C ... +60 °C)		
2A1 0.20mA S.E. (without connector)	753-465	1
2A1 4.20mA S.E. (without connector)	753-466	1
¹⁾ Also available individually		
²⁾ Data format for S5 control with FB 251		

Accessories	Item No.	Pack. Unit
753 Series Connectors	753-110	25
Coding elements	753-150	100
Miniature WSB Quick marking system		
plain	248-501	5
with marking	see pages 352 ... 353	

Approvals	
Also see "Approvals Overview" in Section 1	
Conformity marking	CE
UL 508	
ANSI/ISA 12.12.01	Class I, Div. 2, Grp. ABCD, T4
EN 60079-0, -15	I M2 / II 3 GD Ex nA IIC T4
EN 61241-0, -1	
EN 60079-0, -11, -15	I M2 Ex d I
	II 3 G Ex nA IIC T4
EN 61241-0, -1, -11	II 3 D Ex ID A22 IP6X T135 °C
	* Permissible operating temperature: 0 °C ... +60 °C

Technical Data	
Number of inputs	2
Power supply	via system voltage DC/DC
Current consumption (internal)	75 mA
Input voltage (max.)	10V
Signal current	0 - 20mA (750-465, 753-465)
	4 - 20mA (750-466, 753-466)
Input resistance	< 220Ω / 20mA
Resolution	12 bits
Conversion time (typ.)	2 ms
Measuring error (25 °C)	< ± 0.2 % of the full scale value
Temperature coefficient	< ± 0.01 % / K of the full scale value
Isolation	500 V system/supply
Bit width	2 x 16 bits data
	2 x 8 bits control/status (optional)
Wire connection	CAGE CLAMP [®]
Cross sections	0.08 mm ² ... 2.5 mm ² / AWG 28 ... 14
Stripped lengths, 750/753 Series	8 ... 9 mm / 0.33 in
	9 ... 10 mm / 0.37 in
Width	12 mm
Weight	52.5 g
EMC: CE - immunity to interference	ecc. to EN 61000-6-2 (2005)
EMC: CE - emission of interference	ecc. to EN 61000-6-4 (2007)

WAGO Kontakttechnik GmbH & Co. KG
Subject to design changes

18.05.2012

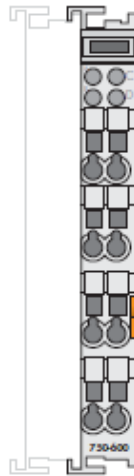
Postfach 2880 - D-32385 Minden
Hansastr. 27 - D-32423 Minden

Phone: +49(0)571/887-0
Fax: +49(0)571/887-169

E-Mail: info@wago.com
www.wago.com

750-600

End Module



After the fieldbus node is assembled with the correct buscoupler and selected I/O modules, the end module is snapped onto the assembly.

It completes the internal data circuit and ensures correct data flow.

[illegible]

Subject to design changes
10.09.2008

WAGO Kontakttechnik GmbH & Co. KG

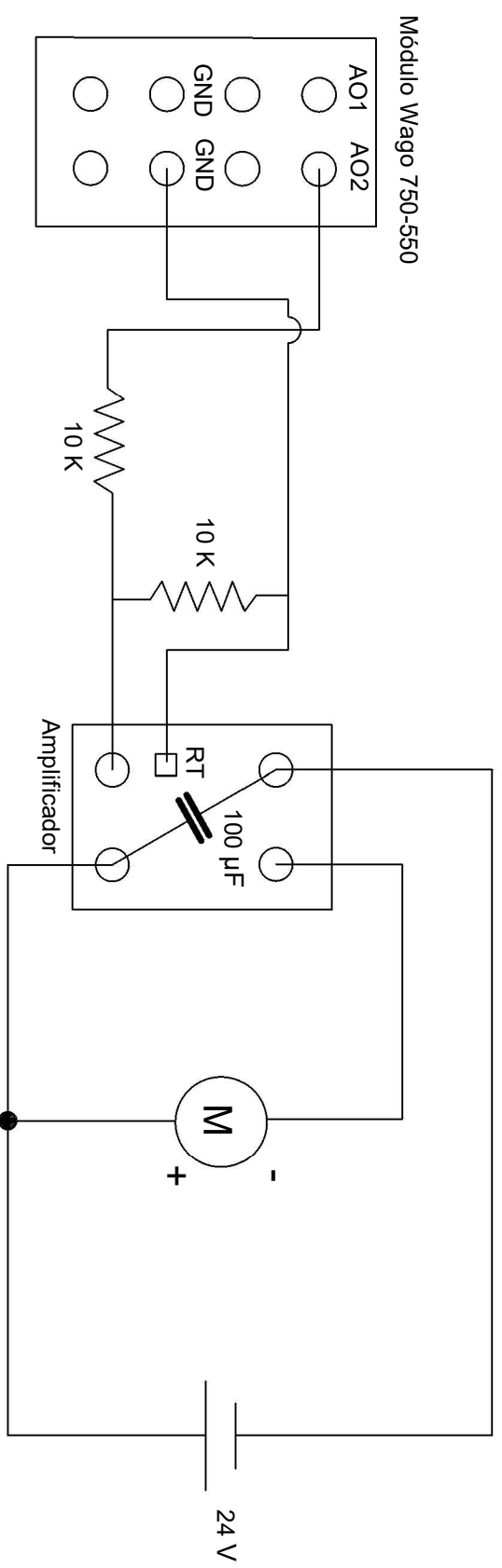
Postfach 2880 • D-32385 Minden Tel.: +49(0)571/887-0 E-Mail: info@wago.com
Hansastr. 27 • D-32423 Minden Fax: +49(0)571/887-169 www.wago.com


PLANOS

Índice Planos

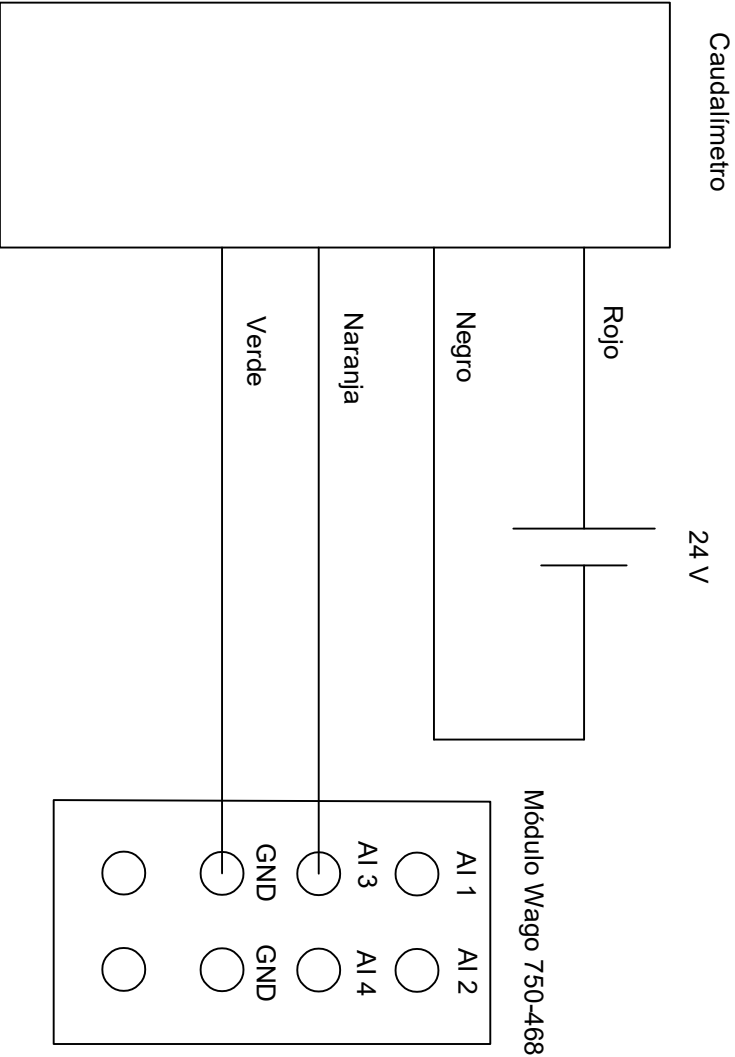
PLANO DE CONEXIONADO DE LAS BOMBAS	115
PLANO DE CONEXIONADO DE LOS CAUDALÍMETROS	119
PLANO DE CONEXIONADO DE LAS ELECTROVÁLVULAS	123
PLANO DE CONEXIONADO DEL SENSOR DE NIVEL ELOBAU	127
PLANO DE CONEXIONADO DEL SENSOR DE NIVEL ENDRESS HAUSER....	131


PLANO DE CONEXIONADO DE LAS BOMBAS



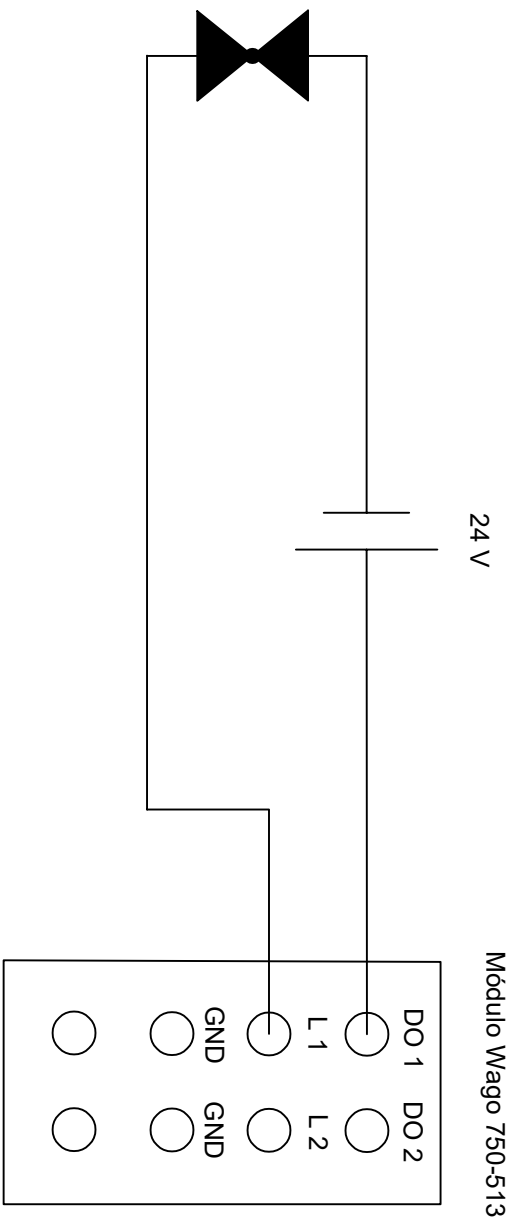
A4	Título: Plano conexionado de las bombas	Plano nº 1
	Autor: Johnata Lorenzo Fernandes	Fecha: 15/10/19


PLANO DE CONEXIONADO DE LOS CAUDALÍMETROS



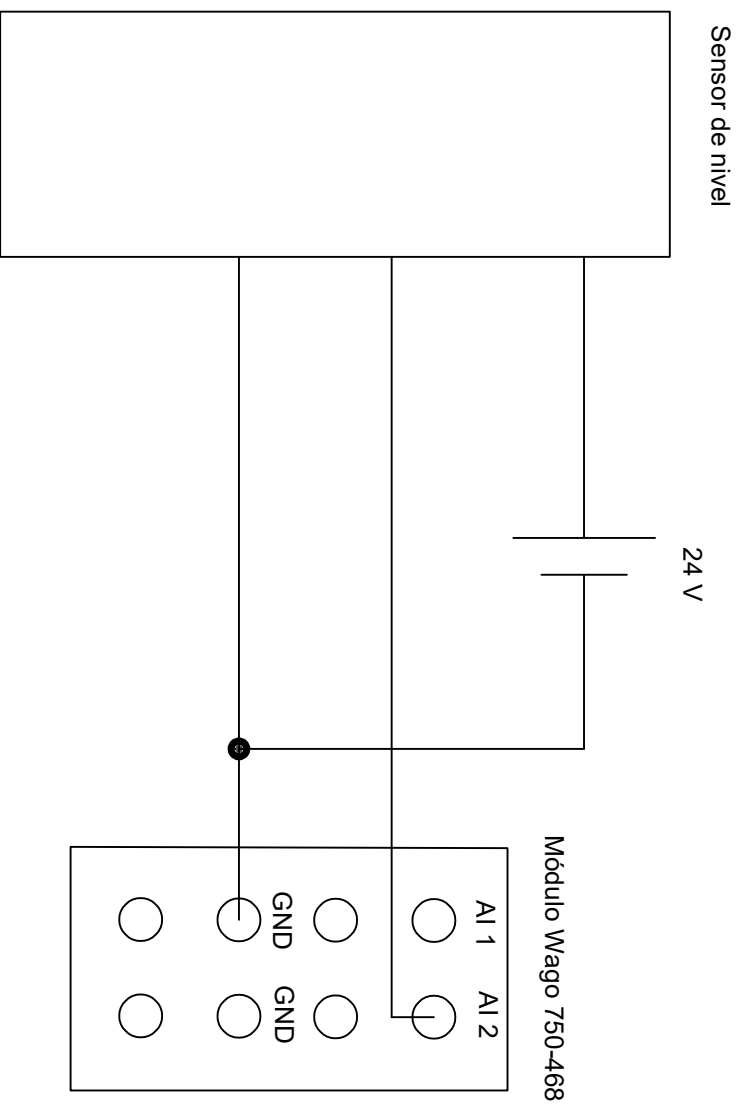
A4	Título: Plano conexionado de los caudalímetros	Plano nº 2
	Autor: Johnata Lorenzo Fernandes	Fecha: 15/10/19


PLANO DE CONEXIONADO DE LAS ELECTROVÁLVULAS



A4	Título: Plano conexionado de las electroválvulas	Plano nº 3
	Autor: Johnata Lorenzo Fernandes	Fecha: 15/10/19

PLANO DE CONEXIONADO DEL SENSOR DE NIVEL ELOBAU

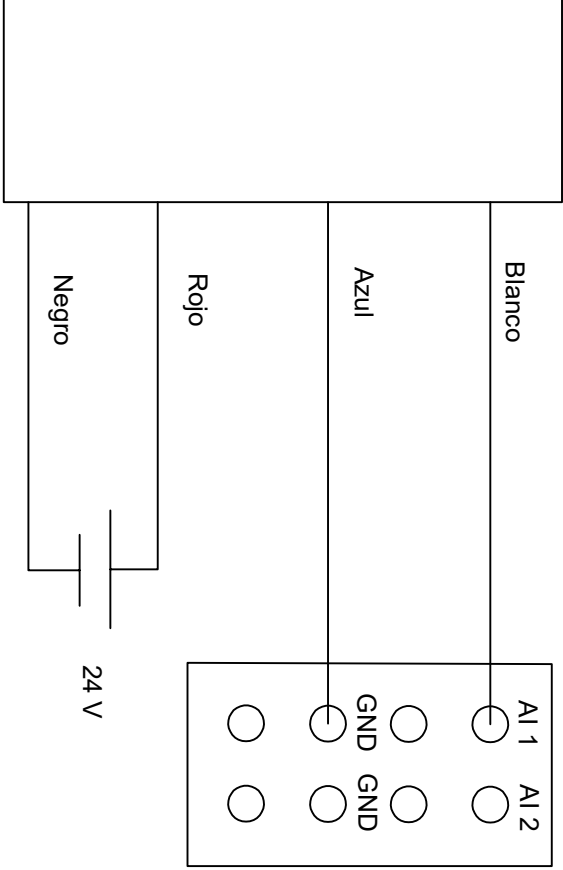



A4	Título: Plano conexionado de sensor Elobau	Plano nº 4
	Autor: Johnata Lorenzo Fernandes	Fecha: 15/10/19

PLANO DE CONEXIONADO DEL SENSOR DE NIVEL ENDRESS HAUSER

Sensor de nivel

Módulo Wago 750-466



A4	Título: Plano de conexionado de sensor Endress Hauser	Plano nº 5
	Autor: Johnata Lorenzo Fernandes	Fecha: 15/10/19

PLIEGO DE CONDICIONES

Índice Pliego de Condiciones

1. REQUISITOS DE LOS EQUIPOS	138
1.1. ORDENADOR.....	138
1.2. SOFTWARE DE PROGRAMACIÓN DEL PC	138
1.3. PLC.....	138
1.4. SOFTWARE DE PROGRAMACIÓN DEL PLC.....	138
1.5. BOMBAS	138
1.6. AMPLIFICADORES	139
1.7. CAUDALÍMETROS.....	139
1.8. SENSORES DE NIVEL.....	139
2. NORMAS DE SEGURIDAD.....	139

1 REQUISITOS DE LOS EQUIPOS

1.1 ORDENADOR

A continuación, se presentan los requisitos mínimos que debe tener el ordenador a usar para poder hacer uso de los programas necesarios en este proyecto:

- Sistema Operativo: Windows 7 SP1 o superior.
- Procesador (CPU): Cualquier procesador x86-64 Intel o AMD.
- Disco Duro: 30 GB de espacio disponible,
- Memoria RAM: 4GB
- Tarjeta gráfica (GPU): 512MB o superior.
- Actualizado a la última versión de Java.

1.2 SOFTWARE DE PROGRAMACIÓN DEL PC

Es necesario del uso del programa MATLAB en su versión 2018 o superior, puesto que a partir de esta se encuentra disponible la librería MODBUS que permite la comunicación con el PLC.

1.3 PLC

El PCL a utilizar debe ser de la marca WAGO puesto que el código desarrollado en este proyecto se adapta a características específicas de su hardware. En caso contrario podrían producirse errores en la ejecución del código y la comunicación MODBUS.

1.4 SOFTWARE DE PROGRAMACIÓN DEL PLC

El software de programación del PLC será Codesys ya que es el software compatible con los productos de la marca WAGO. En el caso de utilizar el PLC 750-741 como en este proyecto, se usará la versión Codesys V2.3 que es la que contiene los drivers adecuados para este. Si se utilizan otros modelos se deberá consultar el manual de usuario y seguir sus instrucciones.

1.5 BOMBAS

Las bombas a utilizar han de ser sumergibles para poder trasegar el agua desde el fondo de los tanques y no verse influenciadas por el volumen de agua presente en estos. Deben poder bombear un caudal mínimo de 2 litros por minuto y estar alimentados a 24V o menos.

1.6 AMPLIFICADORES

Los amplificadores deberán soportar tensiones de 24V y corrientes máximas de 2A de forma que sea compatible y pueda ser utilizado con seguridad junto con las bombas. También dispondrá de una entrada analógica para el control de los motores de continua.

1.7 CAUDALÍMETROS

Deberán tener un rango de medición entre 0-10 litros por minuto, una baja caída de presión hidráulica y una salida analógica que comunique los datos medidos. La resolución mínima exigida es de 0.001 litros por minuto.

1.8 SENSORES DE NIVEL

Los sensores de nivel tendrán un rango de medición mínima de 50 cm de forma que se garantice un mínimo nivel de evolución del nivel de los tanques durante las pruebas. Este sensor también deberá contar con una salida analógica para la comunicación de los datos medidos. Su resolución mínima será de 0.001 litros.

2 NORMAS DE SEGURIDAD

A la hora de realizar el conexionado eléctrico de los distintos elementos del sistema con la alimentación, este se hará en todo momento con la fuente de alimentación o el interruptor correspondiente apagado.

Se mantendrá el lugar de trabajo limpio y libre de obstáculos que puedan producir caídas accidentales. Por otra parte, al trabajar con agua en este proyecto, se tendrá especial cuidado de mantener el entorno seco de forma que se evite por una parte posibles resbalones y por otra los contactos eléctricos en mojado.

PRESUPUESTO

En este apartado se detallan los costes de la realización de este proyecto dividido entre coste material y mano de obra.

1 COSTE MATERIAL

Componentes	Unidades	Precio unitario (€)	Precio total (€)
Bombas	4	16,9	67,6
Electroválvulas	4	21,24	84,96
Caudalímetros	4	210	840
Sensores de nivel	2	370	740
Amplificadores	4	57,18	228,72
Componentes aux.	-	-	20
Total			1981,28

Tabla 8. Coste de los componentes de la maqueta hidráulica

Componentes	Unidades	Precio unitario (€)	Precio total (€)
PLC WAGO 750-841	1	340,82	340,82
Modulo 750-468	2	37,53	75,06
Modulo 750-550	2	111,6	223,2
Modulo 750-513	2	45,9	91,8
Modulo 750-466	1	62,8	62,8
Modulo 750-600	1	23,68	23,68
Total			817,36

Tabla 9. Coste de los componentes sistema de control

2 COSTE DE LA MANO DE OBRA

Tarea Realizada	Tiempo Invertido
Diseño y montaje hidráulico	30
Diseño y montaje eléctrico	30
Desarrollo del programa en el PLC	90
Desarrollo del programa en Matlab	150
Total	300

Tabla 10. Tiempo invertido en el proyecto por cada tarea

Horas de personal	Horas	Precio unitario (€)	Precio total (€)
Ingeniero	300	30	9000

Tabla 11. Coste horas personal

El coste total del proyecto suma un total de 11.798,64 euros.